2013

Wireless sensor network for precision agriculture: Design, Performance Modeling and Evaluation, and Node Localization

Herman Singh Sahota
Iowa State University

Wireless sensor network for precision agriculture:

Design, Performance Modeling and Evaluation, and Node Localization

by

Herman Singh Sahota

A dissertation submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Major: Computer Engineering

Program of Study Committee:

Ratnesh Kumar, Co-major Professor

Ahmed E Kamal, Co-major Professor

Zhengyan Zhu

Robert Weber

Amy Kaleita-Forbes

Iowa State University

Ames, Iowa

2013

DEDICATION

To my father.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENTS

I take this opportunity to express my gratitude to everyone that helped me during the various aspects of conducting research and the writing of this thesis. I thank Dr. Ratnesh Kumar for giving me this opportunity to work with him and Dr. Ahmed Kamal for agreeing to be a co-major professor. I acknowledge both for their guidance, patience and support. Their insights have made this entire experience pleasurable and thought provoking. I thank my other committee members Dr. Amy Kaleita-Forbes, Dr. Robert Weber and Dr. Zhengyan Zhu for generously sharing their time and ideas and contributing to this work. I am grateful to Dr. Stuart Birrell for his active part in the field experiments and our group seminars.

I acknowledge the help of my friend Dr. Lucas Beverlin for spending useful brainstorming hours with me. I thank Dr. Lucien Ouedraogo, Dr. Jing Huang and Dr. Saayan Mitra for their feedback and help with the project. I am grateful to my good friends Awanish Tiwari, Rajiv Singhal, Dr. Matthias Geissbühler, Jason Erbskorn and Sheng Ly for their positive influence on me throughout the years.

I am highly grateful to my parents for their loving support in all my endeavors and being the support pillars of my life.

This work was supported in part by the National Science Foundation under the grants CNS-0626822, NSF-ECS-0601570, NSF-ECCS-0801763, NSF-CCF-081141, and NSF-ECCS-0926029.

ABSTRACT

The use of wireless sensor networks is essential for the implementation of information and control technologies in precision agriculture. In this thesis, we address the challenges associated with the design of such a network system. We present our design of the network stack for a wireless sensor network used for a precision agriculture application where sensors periodically collect environmental data from spatially distributed locations in the farm-field. The physical (PHY) layer in our network allows multiple power modes in both receive and transmit operations for the purpose of achieving energy savings. We design our medium access control (MAC) layer which uses these multiple power modes to save energy during the wake-up synchronization phase. The network layer is designed to custom fit the needs of the application, namely, reliable collection of data and minimization of the energy consumption. The design of various protocol layers involves a cross-layer design strategy. We present analytical models and simulation studies to compare the energy consumption of our MAC protocol with that of the popular duty-cycle based S-MAC protocol and show that our protocol has better energy efficiency as well as latency in a periodic data collection application operating over a multi-hop network of sensor nodes.

We also study the problem of sensor node localization for a hybrid wireless sensor network, with nodes located both underground (sensor nodes) and above-ground (satellite nodes). We consider two types of ranging measurements (received signal strength and time of arrival) from unmodulated signals transmitted between neighboring sensor nodes and between satellite nodes and sensor nodes. The problems are formulated with the goal of parameter estimation of the joint distribution of the received signal strength and time of arrival of the received signals. First, we arrive at power fading models for various communication scenarios in our network to model the received signal strength in terms of the propagation distance and hence, the participating nodes' location coordinates. We account for the various signal degradation

effects such as fading, reflection, transmission, and interference between two signals arriving along different paths. With the same goal, we derive statistical models for the measured time of arrival with the parameters governed by the sensor nodes' location coordinates. The probability distribution of the detected time of arrival of a signal is derived based on rigorous statistical analysis. Then, we formulate maximum likelihood optimization problems to estimate the nodes' location coordinates using the derived statistical models. The results are validated through the implementation of the proposed sensor localization approach in Python using the SciPy optimization package. We also present a sensitivity analysis of the estimates with respect to the soil complex permittivity and magnetic permeability.

The contributions of this work are threefold. We present the system design of a wireless sensor network for use in a large scale deployable periodic data collection application. Next, we develop a thorough performance evaluation of the energy efficiency, throughput and latency of the system and compare with a traditional duty cycle based approach. Finally, we formulate maximum likelihood estimation based frameworks involving received signal power as well as latency measurements to solve the problem of sensor node localization based on relatively cheaper received signal strength measurements and more accurate time of arrival measurements for nodes deployed in multiple physical media (air and soil), and accounting for multi-path effects, signal loss and delays, and Gaussian and Rician fading.

CHAPTER 1. INTRODUCTION AND PROBLEM FORMULATION

Precision agriculture refers to the use of information and control technologies in agriculture. Agricultural inputs such as irrigation, fertilizers, pesticides, etc. are applied in precise quantities as determined by modeling of crop growth patterns to maximize the crop yield and to minimize the impact on the environment. Fertilizer uptake within a field depends on factors such as variability in plant population, nitrogen mineralization from organic matter, water stress, soil properties, pests, etc. These factors vary in space and time. Fertilizers must be applied according to the needs of the crop. If under-applied, the crop yield suffers while if over-applied, contamination of ground and surface water resources may take place leading to issues such as hypoxia. Furthermore, there is a considerable energy cost associated with the production of nitrogen based fertilizers. Also, understanding the process of carbon sequestration, which is thought of as a method to control global warming, could be aided by the data collection from soil sensing and its further analysis.

Traditionally, precision agriculture had been confined to using manual data collection and control techniques. While being unscalable the traditional method may also introduce unnecessary interference to crop growth. Sensor networks reduce person-hour and heavy machinery costs. Remote sensing can monitor large areas of land for long term but the data resolution is much lower and computation costs are high. In order to apply precision agriculture to the production scales, it requires the collection of higher resolution spatio-temporal data than is currently available by methods such as remote-sensing, laboratory tests, etc. The need is also to automate the collection of data to which sensor networks provide a promising solution. Data about soil characteristics such as nitrate concentration, moisture content, etc. can be sensed by densely scattered sensor nodes and automatically collected using wireless networking. Such sensor networks must be deployed below the ground so that they do not interfere with the

overground agricultural activities.

Wireless sensor networks are networks of tiny computing devices, called sensor nodes, integrated with micro-sensing instruments and radio transceivers. With recent developments in the areas of MEMS (Micro-Electro-Mechanical-Systems), computing power, and transceiver design, researchers have envisaged the use of sensor networks consisting of hundreds of nodes spanning a large geographical area. Each sensor node senses and locally stores data corresponding to its coverage area. Wireless communication capability provided by the radio transceiver allows the sensor nodes to transmit and receive data from other nodes within their radio coverage range. Data from large areas of fields can be collected at a fine resolution and analyzed using a wireless sensor network. Many applications have been conceived based on this paradigm. Precision agriculture has also received its due attention. Miniature sensor-boards integrated with radio communication abilities collect data from vast farm fields at a central processing station where appropriate decision making algorithms are run to apply control inputs to the farm fields based on the feedback received from the analysis of the collected data. In this chapter, we start with a collection of the recent work on the application of wireless sensor networks in precision agriculture.

1.1 Related work on wireless sensor networks and precision agriculture

In [2], authors explore the use of wireless sensor networks in an agricultural production environment by studying the needs and priorities of the workers in a vineyard using ethnographic research methods. The work also involved the deployment of a test wireless sensor network of 18 nodes for a period of a few weeks at a vineyard site in the state of Oregon, USA. The use of static as well as mobile sensor nodes is explored. The authors consider questions concerning the data to be collected, its frequency, computational analysis and presentation to the user. An extension of the work is presented in [3] in which they design and deploy a large scale multi-hop wireless sensor network in a vineyard distributed across one hectare area. The network consists of a fixed topology of backbone nodes and leaf nodes. Duty cycling of the radio is used to conserve energy. Backbone nodes operate at a higher duty cycle of 20% while leaf nodes operate at a duty cycle of 3%. Transmission power control is used to allow a larger transmission range

for the backbone nodes. At the application level, the authors explore Nyquist frequency in space and time of temperature variation. They also present results on the battery lifetime of different node types in the topology and packet loss in field experiments.

Researchers in [4] present an implementation of a multi-hop wireless sensor network deployed in a bayberry greenhouse that collects temperature, humidity and light intensity measurements at a central gateway node which, in turn, transmits to a remote server via General Packet Radio Service (GPRS) [5]. Alarm service and micro-environment status information to the user is provided through integration with Google Maps [6]. The system is implemented on TelosB mote hardware [7], designed with the goal of energy efficiency keeping in mind the requirements of research and experimentation, developed at University of California at Berkeley. In order to achieve energy efficiency, a simple duty-cycle based MAC protocol (similar to S-MAC [8]) is used, where the nodes switch off their transceivers and back on again periodically. However, the sleep-active duty cycles of the nodes may drift due to the randomness inherent in the crystal oscillators of the nodes' clocks (owing to various factors such as temperature, humidity, manufacturing artifacts, etc). To correct the clock drifts, FTSP (Flooding Time Synchronization Protocol) [9] is used to synchronize the active and sleep cycles of the entire network. In [10], authors present results from a similar deployment in an eggplant field and implementation of a fire detection system. The authors stress on the importance of energy efficiency and recovery in the event of node power failures. In [11], authors deploy a wireless sensor network to monitor the micro-climate of a green house. The stored data is analyzed using linear regression and DIF analysis [12] [1] to manage the conditions in the greenhouse to be close to the optimal for plant growth.

In [14], authors present an incremental node deployment scheme to address the specific crop monitoring problem of estimation of the leaf area index (LAI) for a large farmland. LAI is used represent the upper leaf coverage in crop canopy per unit land area. The goal is to reduce the deployment cost while achieving the desired precision in the estimation of the global LAI value. The network senses light intensity at various points in the field and uses a quasi-integral

[1] DIF is a greenhouse technique involving temperature control to control plant internode length and thus elongation rates [13].

algorithm for the estimation.

In [1], authors present experiences with connecting a wireless sensor network for precision agriculture with the Internet of Things, to allow connections between agronomists, farmers and crops. Gateway nodes capable of TCP/IP communication are used to integrate the sensor network with the Internet. A simple access authentication method is included in the protocol to allow secure access to the data. The authors aim to provide agronomists with the means to gain a better understanding of plant growth models by utilizing data from farmlands and greenhouses that are owned by other agronomists and farmers in order to breed and improve specific crop varieties. Similarly, farmers can gain tremendously by such an instrument by gathering data from farm fields in the broader geographic region and be better informed about climate variability, pests and disease outbreak. Figure 1.1 (taken from [1]) shows the proposed system architecture. Similarly, many other researchers have been working on the application of wireless sensor networks in precision agriculture, as seen in [15], [16], [17], [18], [19], [20]. In the next section, we look at the requirements imposed on a wireless sensor network that is to be used for precision agriculture applications, based on our literature survey.

1.2 Requirements and challenges

Wireless sensor networks must satisfy certain practical requirements in order to be useful for precision agriculture. The following are some key requirements.

- Sensor nodes must not interfere with the agricultural practices. The deployment of the nodes must be such that the regular agricultural practices of tillage, irrigation, fertilizer and pesticide application are not interrupted in any manner.

- The sensor network must be energy efficient so that the battery operated nodes last for periods ranging over crop season(s). It was observed in [3] that the backbone nodes deplete their battery energy at a much faster rate than the leaf nodes even though the radio duty cycle was not at 100%. Hence, we notice that reducing the radio duty cycle is not enough. We must design the entire protocol stack, especially the MAC and network layers, to be energy efficient.

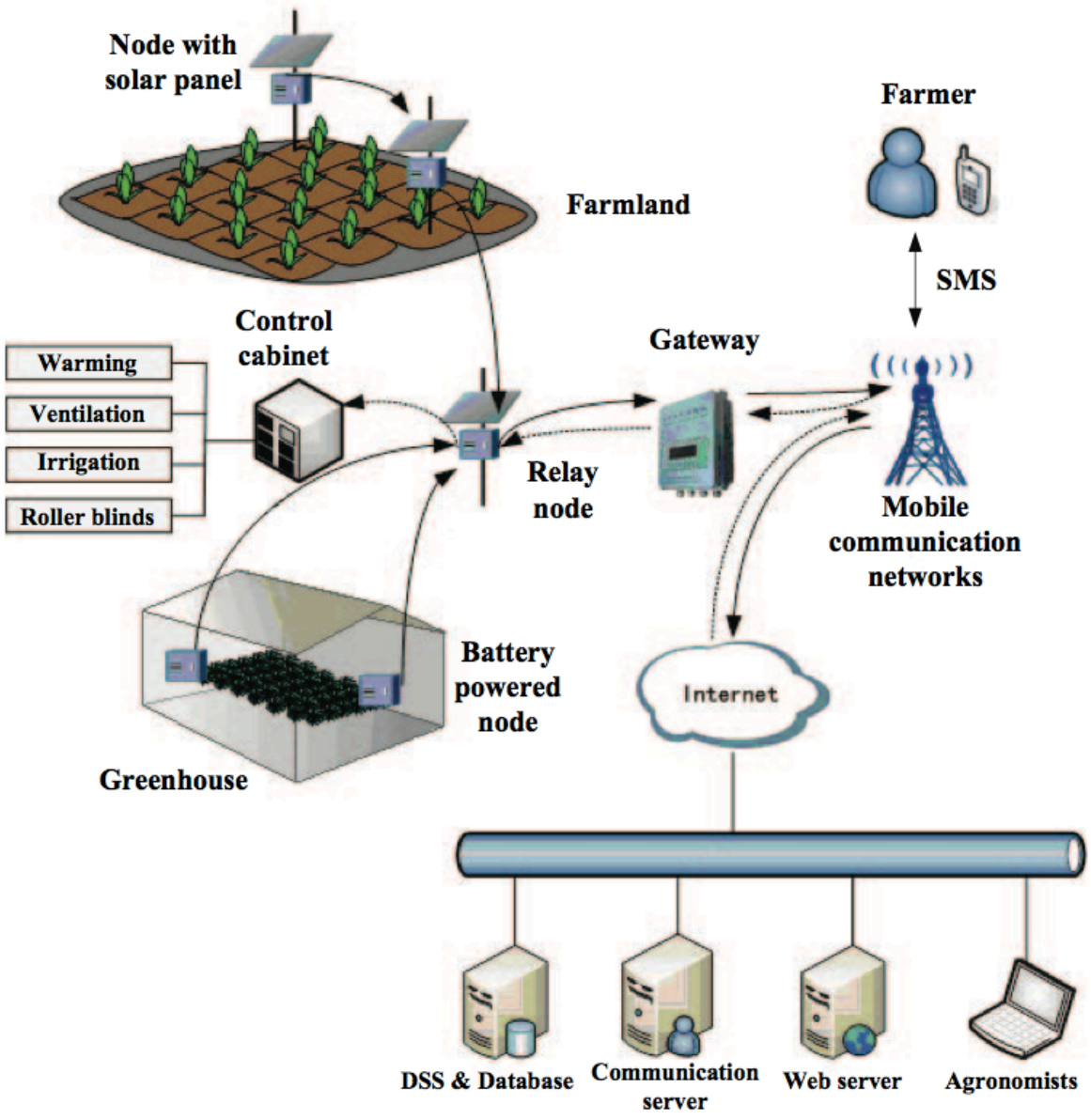


Figure 1.1 System architecture of integration of precision agriculture WSN with Internet of Things [1].

- Most applications of wireless sensor networks in precision agriculture are envisaged to be of a data monitoring nature in that the nodes spread over the farm field collect spatio-temporal data periodically at a central processing station. The data is analyzed and fed to crop growth models to identify the optimal values of the agricultural inputs to be fed to the control equipment(s). In order for the models to be accurate in prediction, the collected data must have a high spatio-temporal resolution. This is especially a challenge in an outdoor environment when channel conditions are prone to disturbances in the environment. In [3], authors report packet loss rates of about 77% in field conditions compared to 92% in lab performance. Thus, reliability is a key requirement.

- The application of control inputs in agriculture is time-sensitive but without a hard deadline. Hence, we envision a loose upper bound on the data propagation latency.

- Traditionally, precision agriculture farm field shapes and sizes have been guided by factors such as the ability to differentially sense and manage different portions of the field and the patterns in which the agricultural equipment moves. It is hoped that the use of wireless sensor networks will allow more flexible field sizes and shapes. For example, the sensed area can range from a small green house to a field of area of the order of hectares. Thus, scalability of the system is a desired requirement.

- In order for commercial success, the system must be cost effective; that is, the profits obtained from the increased and improved yield must be higher than the installation costs.

- Sensor networks must be remotely manageable so that the actions such as diagnosis, reconfiguration and software upgrade can be performed easily.

- With the advent of the Internet of Things, the data and analysis from the use of precision agriculture must be pervasively available which would require interoperability with other technologies such a Wi-fi, GSM, etc.

- The ability to automatically identify the location coordinates of the nodes is desirable. Such information can be used to feed the network layer algorithms and assign meaningful

representation to the collected data.

Wireless sensor networks have been the subject of researchers' interest over the past decade. Since the inception of TinyOS (an operating system for embedded devices), at University of California at Berkeley in 2000, sensor networks have attracted researchers in various sub-disciplines such as protocol design (medium access control and network layers), sensor design (micro-electro-mechanical systems), transceiver design, application design (e.g. sensor node localization), etc. Various application scenarios using sensor networks have been envisioned, which can be broadly divided into two categories: event detection and periodic data collection. The subject of this thesis is a periodic data collection application used in the context of precision agriculture. In the next section we present a formulation of the problems that we address in this thesis.

1.3 Problem formulation

We study the problem of network design for a wireless sensor network used in a data collection application in precision agriculture. Specifically, we seek solutions to the design of MAC/PHY and network layers for such an application. Next, to quantify the effectiveness of the developed protocols, we formulate methods to scientifically evaluate the performance of the network in terms of different metrics of interest. Finally, we look at the problem of node localization.

1.3.1 MAC/PHY layer design

MAC protocol allows the nodes to access the wireless channel to communicate with the set of nodes in its radio communication range. Sensor nodes in an application such as precision agriculture are expected to remain unattended for months, while still being required to function. Sensor nodes are battery powered and battery replacement is a costly operation. Thus, it is desirable to have a long node lifetime to minimize the frequency of such maintenance operations. In addition, due to the nature of the outdoor environment the channel characteristics are highly unstable causing packet losses. We seek the design of a MAC layer protocol that minimizes

the energy consumption in the transceiver module, while allowing reliable data communication. Chapter 2 addresses this topic.

1.3.2 Network layer design

Sensor nodes have a limited radio transmission range. In order to conserve energy, it is desirable to transmit at the lowest possible energy level to maintain communication with the nodes in the immediate geographical neighborhood only. This necessitates the use of multi-hop communication in order to allow data collection coverage over large farmlands. It is not only necessary to conserve the energy at individual nodes but also to ensure that the energy consumption is balanced throughout the entire sensor field. This is necessary so that the network does not get partitioned due to a group of nodes failing. Node level transmission and reception schedule is also computed at this layer to minimize the frequency of wake-up synchronizations and data communications. We study the design of an energy efficient and load balanced routing strategy for our sensor network application. Chapter 2 also addresses this topic.

1.3.3 Performance evaluation

We require rigorous methods to test the performance of the proposed communication protocols under the given application scenario. This may be needed to calibrate the protocol parameters to optimize the performance as well as to compare different protocols on the basis of their performance. Since a wireless sensor network used in such applications may consist of hundreds of nodes, it is not readily feasible to quantify the performance metrics (energy efficiency, throughput, latency, etc.) by actual field deployments. Mathematical modeling and simulation techniques are used in such cases. We present our analytical model for performance evaluation for our MAC and Network layer protocols along with that for a well known duty cycle based MAC protocol (S-MAC) under the settings of our application in Chapter 3.

1.3.4 Node localization

Lastly, we study the problem of sensor node localization to determine the coordinates of the sensor nodes. This is useful to design efficient routing protocols as well as to assign meaning to the data collected by the sensor nodes. We formulate localization as a maximum likelihood optimization of the parameters of the distribution of the two types of ranging measurements (received signal strength and time of arrival) used in our application. Chapter 4 presents our work on the sensor node localization problem.

CHAPTER 2. SENSOR NETWORK PROTOCOL DESIGN

2.1 Introduction

In this chapter, we address the problem of designing a sensor network for automated collection of soil data. The protocols are designed to meet the requirements of the application (periodic collection of soil readings from spatially distributed locations) and need to be energy efficient to maximize the network lifetime. Our application requires the collection of soil data over the entire duration of crop season(s) at a sampling frequency of about once an hour and a spatial resolution of about $50 \times 50\ m^2$. Latency is not a strict requirement, but data integrity with a certain confidence has to be guaranteed. The nodes are arranged at rectangular grid points at the required spatial resolution. The sensors integrated with the nodes sense data periodically and at the specified frequency. The sensed data is relayed to the sink node located at one of the corners of the rectangular field.

It is well established that wireless communication is the most energy consuming operation at a sensor node. This chapter motivates the design of a novel physical (PHY) layer and our design for the medium access control (MAC) layer and the network layer with the goal of minimizing the energy consumption of communication. Our approach fits the application at hand and exploits the advantages of a cross-layer design strategy in achieving energy savings.

We begin by introducing the PHY layer in Section 2.4.1. We motivate the design of a radio which operates at different power levels in the transmit mode as well as in the receive mode. The transmit mode has one higher power level mode in addition to the normal two modes of operation of sleep and transmit. Similarly, the receive mode has one intermediate power level mode corresponding to degraded sensitivity in addition to the normal two modes of sleep and receive.

The clocks of the nodes are synchronized periodically to keep them from drifting away from each other. Still, during the period since last synchronization, the clocks of the nodes can exhibit significant drift with respect to each other, and a pair of nodes may not wake up at the same time as scheduled. Hence, a wake-up synchronization phase is an integral part of the communication in such networks. Depending on the implementation of this wake-up synchronization strategy, energy is either wasted at the transmitter side in the form of overemitting (transmission of a message when the destination is not ready to receive) and/or at the receiver side in the form of idle listening.

In the current designs, the wake-up synchronization between a transmitter and a receiver is established using the normal transmission-power and receiver-sensitivity levels. We consider a modified wake-up synchronization scheme in which the receiver saves its battery energy consumption by reducing its own sensitivity level. However this requires increasing the transmission-power level for the transmitter (and hence increasing its battery energy consumption) for successful reception of the wake-up synchronization signal by the receiver. It turns out that this new scheme can save the overall energy consumption during the synchronization phase since the protocol uses a transmission duration that is much shorter than the listening duration [21]. This concept is illustrated through Figure 2.3 and discussed in Section 2.4.

Our MAC layer has been designed to take advantage of these extra power modes, especially in the wake-up synchronization phase. Additionally, we take advantage of the converge-cast nature of sensor network traffic to minimize the number of high energy pings transmitted. This is achieved by making the downstream node wake up all its upstream neighbors with a common ping signal. Existing synchronized and un-synchronized MAC protocols such as S-MAC, T-MAC and WiseMAC establish the individual link communications independently, regardless of whether one of the nodes on the link is common to two or more links. The details of the MAC protocol are described in Section 2.4.2.

The data from sensor nodes is collected periodically, and each data collection period is called a data collection round, or simply a *round*. The routes (Section 2.4.3) from all the sensor nodes to a sink, located at one of the four corners of a field, are computed taking into consideration the remaining battery level of the nodes so as to balance the energy consumption among all

nodes. Note that either of the four corner nodes of a field are capable of functioning as a sink node, and in order to even out the energy consumption across the sensor nodes, we rotate the role of the sink among the four corner nodes after each round. For each round, the routes thus computed determine a certain order in which the nodes must forward their data (e.g. the node furthest from the sink should forward its data first). Adhering to this order, a schedule is created that determines the times at which the various nodes need to communicate with their neighbors in order to forward the data to a designated sink. This scheduling is performed also as part of the network layer (Section 2.4.3). Note that this "node scheduling" is different from packet level scheduling, which is performed at the MAC layer. Also, our approach uses a cross-layered interaction between the physical, MAC and network layers.

The current protocol is geared towards reducing the energy consumption, and our future work plan includes enhancing the protocol to incorporate fault management (both detection and mitigation). In our current design, the lost packets are simplistically dealt with by allowing at most two additional chances for transmission. It is assumed that this will satisfy a desired level of confidence of error free data delivery. However, this can be adjusted based on measured packet error rates.

The following are the contributions of this chapter:

- We consider multiple power modes for the radio and show how it can help design the MAC layer that reduces the energy consumed during the wake-up synchronizations (Section 2.4).
- We develop a routing protocol, and schedule the nodes according to the computed routes so as to balance out the load of communicating sensor data to a sink node, and also to keep the number of wake-up synchronizations to a minimum (Section 2.4.3).
- We develop TOSSIM [22] based simulations of a wireless sensor network, modeling the PHY, MAC and network layers to study the performance of a wireless sensor network of the type described above, and show the resulting energy savings (Section 2.5).
- We present an initial implementation of our MAC protocol on CC1110 SoC with low-power RF transceiver and 8051 MCU from Texas Instruments [23] (Section 2.5).

2.2 Related work

There has been much research on designing energy efficient communication protocols for sensor networks. Energy is wasted during collisions, overhearing, control packet overheads, idle listening, overemitting, etc. TDMA based protocols avoid collisions and overhearing but at the cost of higher overhead in control packets and synchronization. Idle listening is minimized in most protocols (such as S-MAC [8], T-MAC [24], DS-MAC [25], B-MAC [26]) by periodic sleep-active schedules. For example, in S-MAC, the nodes in the same virtual cluster (determined by the communication range of the nodes) share the same sleep-active schedule. The nodes on the boundary of two clusters follow two different sleep-active schedules. T-MAC is a modification of S-MAC. In T-MAC, during the active period, if the node does not detect any activation event for a certain timeout period, it goes back to sleep. The advantage is the reduction in the wastage of energy in the form of idle listening while the drawback is the early sleeping problem which happens because of the broken synchronization between the nodes. DS-MAC adds the dynamic duty cycle feature to S-MAC to decrease the latency. WiseMAC [27] allows a very low duty cycle for the nodes at the cost of longer preambles, causing energy wastage in the form of overemitting. Each data packet is preceded by a long preamble to alert the receiver of the oncoming transmission. The nodes periodically wake up to listen to any activity on the channel. If a node detects a preamble being transmitted during this listen phase, it stays awake to receive the transmission. A time-line of the events in WiseMAC is shown in Figure 2.1. Clearly, there is potentially a large energy wastage in the wake-up synchronization phase in the form of overemitting. To reduce this energy wastage, the sender can learn the sleep schedule of the receivers (through the ACK messages) and dynamically shorten the length of the preamble.

In our application also, clocks drift over a period since the last synchronization was performed, and since no synchronization occurs for long periods (such as between any two rounds which is of the order of hours), the drift can be significant. Hence, our strategy is aimed at improving the energy wastage during the wake-up synchronization phase of the transmission, as discussed in the following sections.

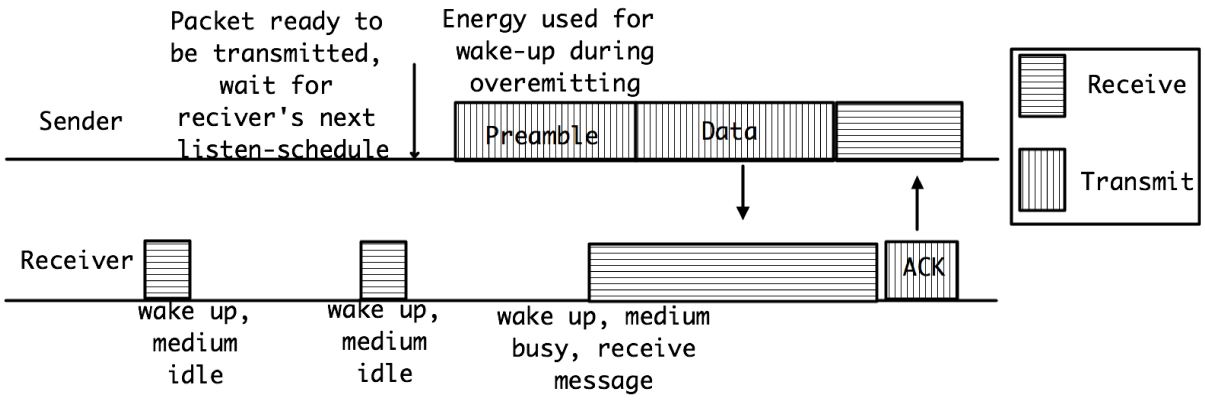


Figure 2.1 Time line of events in WiseMAC.

2.3 Network model

Sensors are deployed in a farm that is divided into regular square fields (see Figure 2.2). Within each field, sensors are deployed in a rectilinear grid form, with equal horizontal and vertical distances of 50 m. Sensor nodes use a transmission power that enables them to reach their horizontally and vertically adjacent nodes. Sensors are deployed such that they are buried in the soil at a depth between 30 cm and 50 cm. This deployment was chosen in order to ensure sensing coverage of the entire field at a resolution required by precision agriculture. Samples should be taken from spatial points in the field where the spatial lag[1] enables the model to predict the moisture and nitrogen distribution. Typically, this value varies from 10 m to 100 m for precision agriculture applications, and it was set to 50 m in our network in order accommodate a maximum transmission range of about 70 m at the chosen frequency of 27 MHz. The sensor nodes can be easily deployed according to this strategy during field operations since farm workers and equipment move in this field in regular patterns, which coincide with the target deployment. Moreover, such a deployment strategy makes it easy to recover the sensor nodes (for battery replacement, etc.).

At the corners of each field, satellite stations are deployed. The satellite stations are battery powered, may be equipped with computing equipment, and can be reached easily for battery replacement. Satellite nodes have a transmission range that can cover all nodes in the field. Communication between sensor nodes in a field and the satellite stations at the corners of the

[1]Spatial lag is the term used for the distance between adjacent sampling points.

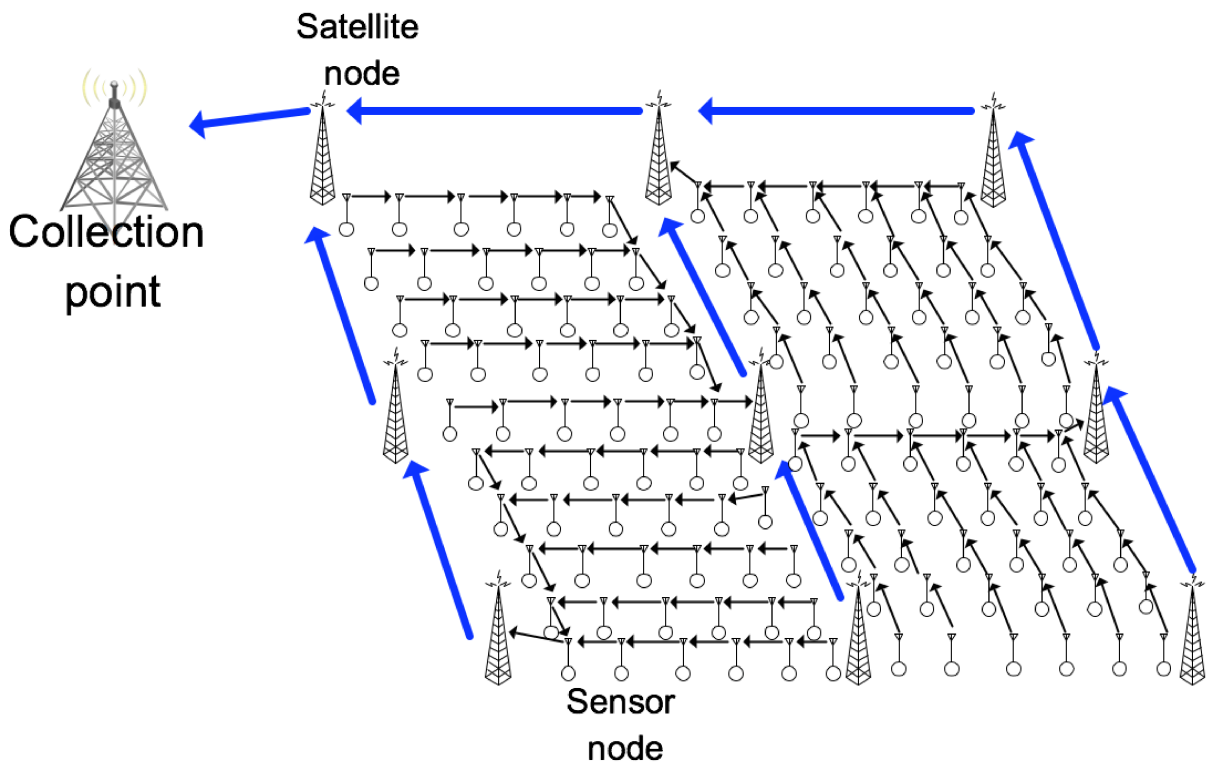


Figure 2.2 Wireless sensor network, showing sensor and satellite station deployment.

field is asymmetric, in the sense that a transmission from a satellite node can reach all sensor nodes in the field, but a transmission from a sensor node reaches adjacent sensor nodes only, and can reach a satellite node if it is close enough (see Figure 2.2). The satellite stations do not perform any sensing operation, but they serve four purposes:

1. they collect information from the sensor nodes in a field,
2. they relay the collected information to a base station where the information is processed,
3. they dispatch routing and scheduling information to the sensors in the field, and
4. they participate in the sensor localization process.

Due to the first two points above, the routing is based on a two-level hierarchy, which can simplify routing and also save energy consumption of the sensor nodes. Moreover, the third point also reduces the complexity of the sensor nodes, reduces their computational requirements, and saves their energy consumption. The fourth point above is the subject of Chapter 4.

The routes within a field are computed and dispatched by the satellite stations. The routing strategy computes routes over sensor nodes in the field such that satellite stations take turns in collecting data, so that energy consumption is balanced. While a static routing strategy can be devised such that the average energy consumption is the same, and is therefore simpler, the alternate routing strategy has the advantage of withstanding failures and achieving a certain level of resilience, without adding any extra measures for this purpose. That is, if part of the network fails, including a single sensor node, under a static routing strategy, certain data measurements, which traverse the failed part, will never be delivered. However, with the alternating routing strategy, a failure can only affect 25% of the measurements, hence leaving 75% deliverable with integrity.

It is to be noted that this network is neither an ad hoc network, nor an infrastructured network. We view this network as a hybrid network, since it combines features from both types.

The above network model is scalable due to the following reasons:

- Since the field size is fixed, and since sensor nodes communicate with adjacent sensor nodes only, increasing the number of fields does not increase energy consumption of sensor nodes.

- With increasing the number of fields, the satellite stations closer to the base stations will have to carry more data, and hence consume more energy. However, the energy consumption is linearly proportional to the number of sensors. If satellite stations perform some processing on the data, energy consumption can be further reduced.

2.4 Energy efficient wake-up and data collection

Energy efficiency is achieved by reducing the energy consumed by the receiver of the wake-up synchronization signal at the PHY/MAC layer level, and by using routing and scheduling to achieve load balancing and keeping the number of internode communications per round to a minimum. Accordingly, we motivate the design of a PHY layer with multiple transmit power and receiver sensitivity levels and discuss the design of MAC and network layers in the following subsections.

Since data sampling is periodic in our application, and the period is long, viz., one hour or longer, sensor nodes sleep most of the time. A node is awakened by its downstream neighbor when it is ready to collect data from the former. The relative synchronization error depends on the sensors' clocks which are controlled by on-board crystal oscillators. Since inexpensive sensor nodes are implemented using inexpensive crystals, the drift in the sensors' clocks can be significant (typically of the order of 30 parts per million). This problem is indeed exacerbated by the fact that the periods over which clocks are synchronized are long (since nodes' sleep periods are long). In the worst case, the receiver of a wake-up signal needs to wake up for a time that is of the order of 4Δ, where Δ is the maximum clock drift during the period since last synchronization (this will be explained below), in order to make sure it can capture the wake-up synchronization signal from its transmitter. Therefore, the energy consumed by the receiver can be significant if Δ is large, which is the case for long sleep periods.

The approach that we take in order to reduce the receivers' energy consumption is based on a trade-off between the energy consumed at the transmitter and that at the receiver of the wake-up synchronization signal: The node transmitting the wake-up signal transmits at a higher power level, while the receivers use less power in the detection of the wake-up signal by degrading their sensitivity. This can be achieved by simplifying the circuitry of the receiver. For example, we can by-pass some amplifier stages, and save on the energy they consume for their operation. The wake-up signal is a signal of very short duration and carries no data. Its sole purpose is to generate an interrupt in the receiver circuit to indicate that it needs to wake up. Accordingly, a wake-up strategy has been proposed by the use of RFID technology in [28].

We term the transmission of the wake-up synchronization signal, which consumes a relatively large amount of power for a very short duration of time, the *ping* signal. Likewise, the mode of operation of the receiver circuit in the degraded sensitivity is termed as *drowsy*. These two modes of the radio are used in the wake-up strategy employed by our MAC protocol to save on the energy consumption during wake-up synchronization.

We call our MAC protocol PD-MAC (letters 'P' and 'D' stand for ping and drowsy, respectively). For the subject of this thesis, we focus our attention on data collection, where multiple sender nodes send their data to a common downstream node. Owing to the convergecast na-

ture of data communication, it makes sense for a receiver node to send a ping pulse to wake up all its upstream sender nodes. The sender nodes must transition to drowsy modes before the receiver node sends out the ping. Once the receiver is awake, it sends a ping pulse to awaken and synchronize all its upstream senders, which then transmit their data one-by-one to the receiver.

Figure 2.4 shows the time-line of events for our MAC protocol for one receiver and two senders. More details can be found in [21].

At a higher level of the protocol stack, namely network layer, we aim at further reducing the energy consumption by seeking load balancing in the computation of routes from the sensor nodes to a sink node. The routes are computed by taking into consideration the remaining battery level of the nodes, and a downstream node is chosen to be a neighbor that has the maximum remaining energy level. Also, the role of the sink is rotated among the four corner nodes after each round of data collection. The schedule for the transmission/reception of messages is computed in accordance with the order of communication as determined by the routes and also to ensure that the number of sender-receiver wake-up synchronizations and internode communications is kept to a minimum.

2.4.1 PHY layer design

We require the radio in a receiver node to be able to operate in a low power receive mode called *drowsy*, and at the same time the radio in a sender node be able to transmit at a high power mode called *ping*, which lasts for a short time, and wakes up the radio of a neighboring node which is in the *drowsy* mode. This trade-off of energy can be effective in saving energy because of the relatively short amount of time that the sender's radio has to spend in the *ping* mode. An RFID based system can be used to implement such as strategy as presented in [28]. Each sensor node is equipped with a tag and a reader. The reader at one node is used to send the *ping* to the tag of a neighboring node to awaken the latter. The tag, on receiving the signal, generates an interrupt which activates the antenna system for regular data communication. Here, the *drowsy* mode is realized by drawing energy from the received signal itself. As mentioned earlier, another way to save energy is to downgrade some parts of the

receiver circuitry.

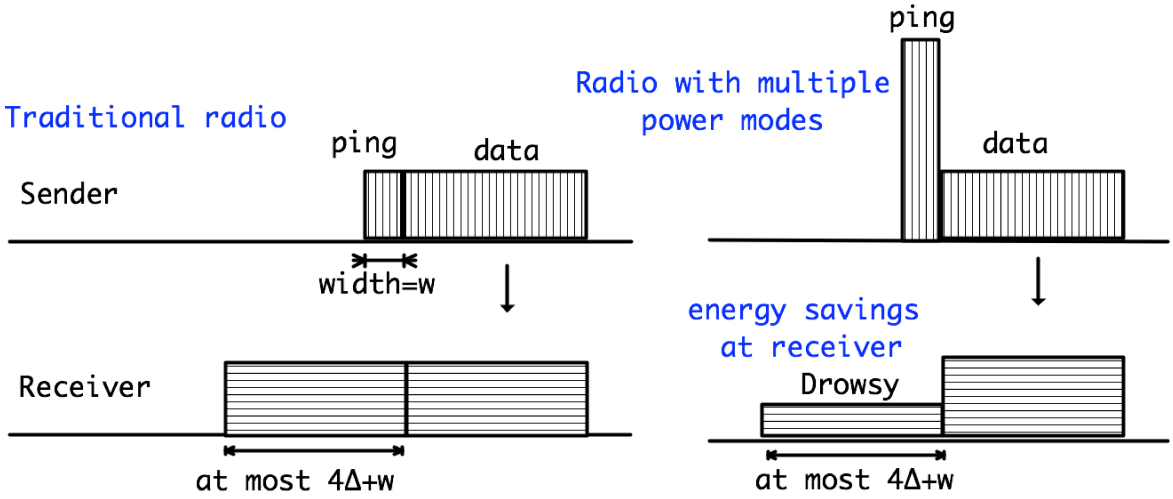


Figure 2.3 Illustration of energy saving at receiver by our scheme.

Figure 2.3 illustrates the energy saving achieved by using the drowsy and ping modes as opposed to the regular transmit and receive levels. Letting Δ denote the duration of clock drift during a sleep-period, w the width of a ping, P_t and P_r the power levels of normal transmit and receive modes, and P_p and P_d the power levels of ping and drowsy modes, the energy saved during a wake-up synchronization can be up to $(4\Delta + w)(P_r - P_d) - w(P_p - P_t)$ (the first term denotes the energy saved at the receiver, and the second term subtracts the extra energy needed for ping). The appearance of the term "4Δ" can be understood as follows: The sender may wake up Δ unit earlier than scheduled, and in order to not miss the ping, the receiver must target waking up 2Δ units earlier than scheduled (so that it will wake up latest by Δ unit prior to sender's scheduled wake-up time). But then the receiver may wake up as early as 3Δ units prior to the sender's scheduled wake-up time whereas the sender may wake up as late as Δ unit after its own scheduled wake-up time, and as a result the receiver may remain drowsy for at most 4Δ units before it starts to witness the ping signal. In order that Δ does not grow unbounded, the clocks need to be synchronized periodically, and in our application this is done at the beginning of each round.

2.4.2 MAC layer design

MAC protocol is responsible for allowing a node to access the physical channel. It uses the interfaces provided by the PHY layer to facilitate the communication among nodes. The two major communication patterns in our sensor network are broadcast (for distributing the routing and schedule information, and a clock reset value to the nodes) and convergecast (for data collection in each round). The routing, schedule, and global clock reset is distributed from one of the satellite nodes located at the corners of the field to all the nodes in the field using a single hop communication. This is possible since the transmission range of the satellite nodes is considerably larger than that of the data collection sensor nodes.

Accordingly, the majority of the functionality of our MAC protocol is aimed at data collection (convergecast communication), where data from all sensors are sent to a single sink node according to a global routing and scheduling information made available to all nodes (that varies from round to round). Owing to the convergecast nature of data communication, it makes sense for a receiver node to send a ping pulse to wake up all its sender nodes. The sender nodes must transition to their drowsy modes before the receiver node sends out the ping.

Once the receiver is awake, it sends a ping pulse to awaken and synchronize all its upstream senders, which then transmit their data one-by-one to the receiver. Figure 2.4 shows the timeline of events for our MAC protocol for one receiver versus two upstream senders, when the first transmission from one of the nodes succeeds while the second node misses the first ping signal.

We have described above the basic idea behind our MAC protocol. Their behaviors are precisely described with the help of finite state machines shown in Figures 2.5(a) and 2.5(b).

2.4.3 Network layer

Multi-hop routes from all the sensor nodes to a sink are computed for each round. The objective is to minimize the energy consumption by way of load-balancing and avoiding more than one successful transmission by each node in each round (since each transmission incurs the

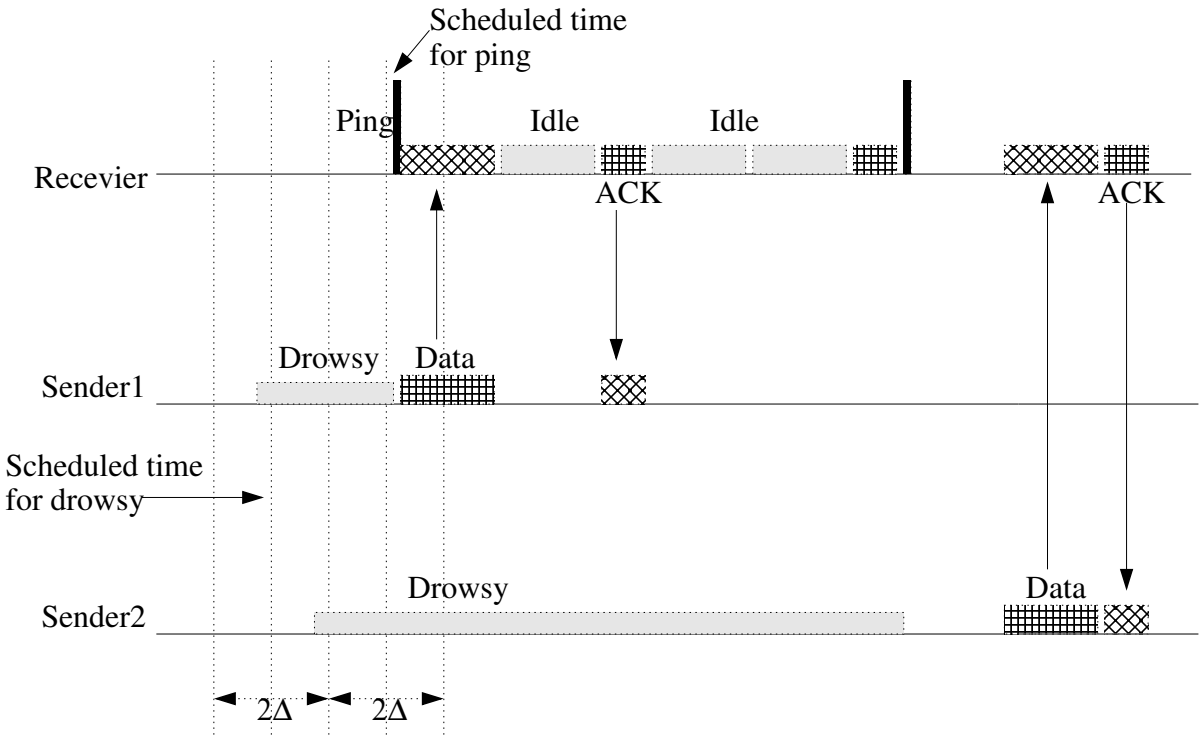


Figure 2.4 Sample time-line of events in our MAC protocol.

additional energy for wake-up synchronization). That is, a node must forward to a downstream node only when it has received the data from all of its upstream nodes to avoid multiple transmissions. In our application, the sensor field is laid out in the form of a square with the nodes placed at the various grid points. In each round, the nodes forward their data to a sink located at one of the four corners of the field. The role of the sink is rotated among the four corners of the field to further balance the energy consumption in the field. Within a round, a node forwards its data to a neighboring node that is closer to the designated sink node. In case of multiple choices for such a neighbor, the data is forwarded to the one with the maximum remaining energy level. This routing strategy helps balance out the energy consumption across the entire sensor field. Figure 2.6 shows a layout of the field and the possible routing patterns for a sequence of four rounds. Such a sequence of 4 rounds, in which each corner node has once acted as a sink, is called a *macro-round*.

The routes determine the order in which nodes must forward their data. The scheduling of the nodes must further be performed to determine the exact timings of events, such as when a

node shall initiate wake-up (by generating ping), when shall a node go into drowsy (to capture a ping), and when shall a node transmit its data together with the data it has collected from its upstream nodes. All such scheduling is also performed within the network layer. A goal of this scheduling is to ensure that the data collection takes place with as few pings as possible. For this, all the senders to a receiver are grouped and awakened together by a single ping from the receiver, and further only those senders are allowed retransmission whose communications failed (at most two retransmissions are allowed in the current scheduling algorithm, but this can be adjusted to meet the requirements of the packet success rate). The routing and the scheduling information is computed by one of the corner satellite stations, and is then distributed to the nodes in the field. Figure 2.7 shows the structure of a routing/scheduling message.

2.5 Simulation results and initial implementation

We have developed a simulation framework for our sensor network in TOSSIM [22], a simulator for TinyOS [29] wireless sensor networks. MAC and PHY layers were implemented in nesC while route and schedule computations were implemented in Python [30]. Python allows a way to control the simulation such as accessing the various nodes in the sensor field, turning them on or off, injecting messages into the network, etc.

Table 2.1 Current drawn as per CC1110 datasheet for various power modes.

Radio mode	Power Level (dBm)	Current drawn (mA)
Transmit	-30 (Normal)	15
Transmit	10 (Ping)	33.5
Receive	-110 (Normal)	19.8
Receive	-107 (Drowsy)	17.8

We model the drift in the crystal oscillator of the nodes using a uniform random variable, the range (Δ) of which depends on the *scheduled time* field value $T_{scheduled}$ of a schedule message delivered to nodes, and the drift in parts per million (ppm) of the crystal oscillator (this is a manufacturer specified value and is know apriori). We used a drift of 30 ppm for an oscillator

of frequency 1 MHz in our simulations. The various required values were computed as follows:

$$\begin{aligned} \Delta &= drift_{ppm} \times T_{schedule} \\ t_{rx}, t_{tx} &\in [T_{schedule} - \Delta, T_{schedule} + \Delta] \\ T_{beginDrowsy} &= t_{rx} - 2\Delta \\ T_{drowsyTimeOut} &= 4\Delta + w \end{aligned} \tag{2.1}$$

The receiver randomly chooses a time t_{rx} in the interval $[T_{schedule} - \Delta, T_{schedule} + \Delta]$ and sends a short ping pulse. A sender node goes into the *drowsy* mode at $T_{beginDrowsy}$ and remains in *drowsy* until it receives a ping from its downstream receiver, or the timer $T_{drowsyTimeOut}$ expires, whichever occurs earlier.

When a sender receives the ping pulse, it gets synchronized with the receiver, and transmits all its accumulated data in its assigned slot (depending on its order relative to that of the other sender nodes of the same receiver as specified in the schedule, see Figure 2.7). The receiver stays in normal-receive mode for a period of $N \times SLOTSIZE$, where N is the number of upstream nodes sending to this receiver and $SLOTSIZE$ is chosen as the maximum size of a data packet expected from any node in the network (taking into consideration the bit-length of all data that can be accumulated at a node).

Figure 2.8 shows the energy consumption for 100 nodes in a 10×10 field for the cases of a traditional radio versus a radio with multiple-power modes and with and without sink node rotation over 10 macro-rounds. It can be seen that for the case of no-rotation there is a large variation in the energy consumed by the nodes because the nodes that are closer to the sink node (shown as the lower id numbers) get more drained than the others. The most even energy consumption among the three cases is observed in the case of the radio with multiple power modes, when load balancing is applied at the network layer and effectively implemented at the scheduling layer.

Figure 2.9 shows the energy consumed by the most energy drained node in fields of different sizes (ranging from 2×2 to 10×10) for traditional radio versus radio with multiple power modes. To make a fair comparison, the other layers (MAC and network) are kept the same in the two cases. The simulation shows the energy consumed by the nodes in 4 macro-rounds (i.e., a

sequence of 16 rounds). It can be seen that the radio with multiple power modes results in higher energy savings.

Moreover, as is clear from Figure 2.10, the radio with multiple power modes results in a more balanced energy consumption profile across the sensor field. This figure shows the difference between the energy consumed by the most energy drained node and the least energy drained node in the field.

Figure 2.11 and Figure 2.12 show the results when the field size is kept constant at 5×5, but the number of macro-rounds is varied from 1 to 10. For each macro-round, the radio with multiple power modes has a better energy efficiency than a traditional radio utilizing the same remaining network stack.

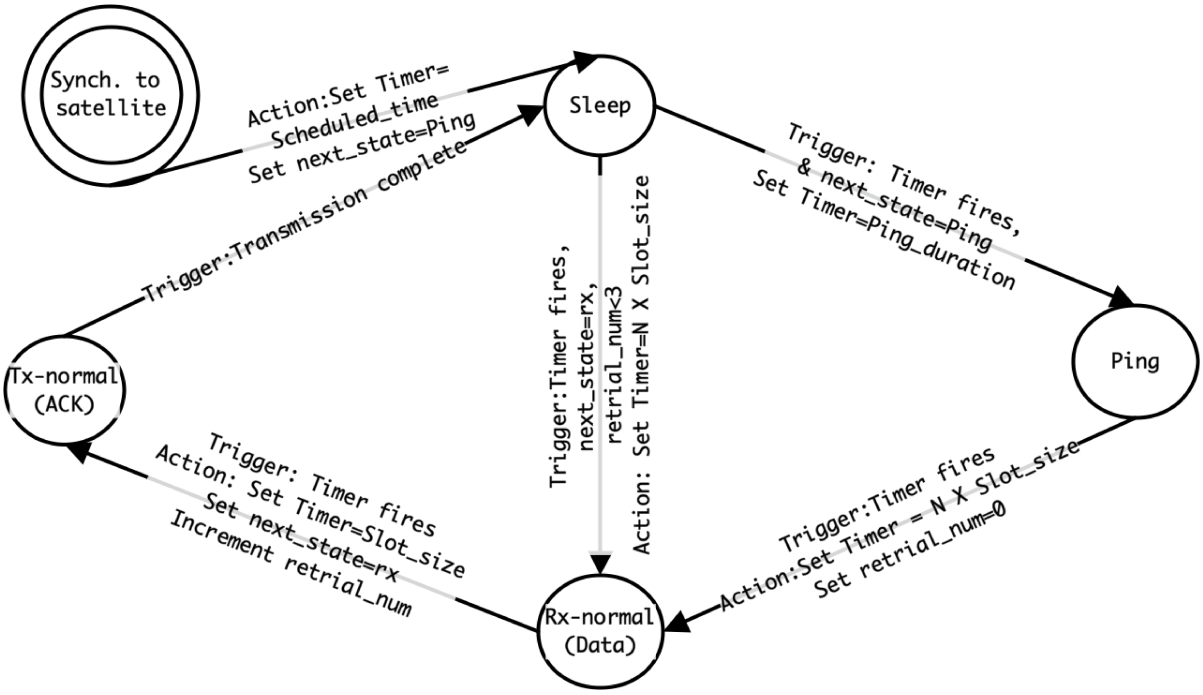
For the proof of concept, we have implemented the protocol for the case of two nodes, i.e., one uplink node and one downlink node. The hardware of the nodes is based on a CC1110 SoC with a low-power RF transceiver and 8051 MCU from Texas Instruments. Specifically, each node is composed of a CC1110 evaluation module (see Figure 2.13(a)) plugged into a SmartRF04 evaluation board (see Figure 2.13(b)), whose LCD, LEDs and buttons are readily available for monitoring and control. The code is written in C and compiled under IAR embedded workbench.

After power on, the two nodes go into the receive mode, and wait to receive a message with the schedule time $T_{schedule}$ from the base station. The uplink node sets its timer to $T_{schedule} - 2 \times \Delta$ while the downlink node sets it to $T_{schedule}$, and then both the nodes transition to sleep mode. The uplink node's timer is set to this value to make sure that it wakes up before the downlink node to receive the ping. See Equation (2.1) to see how the uplink node computes Δ.

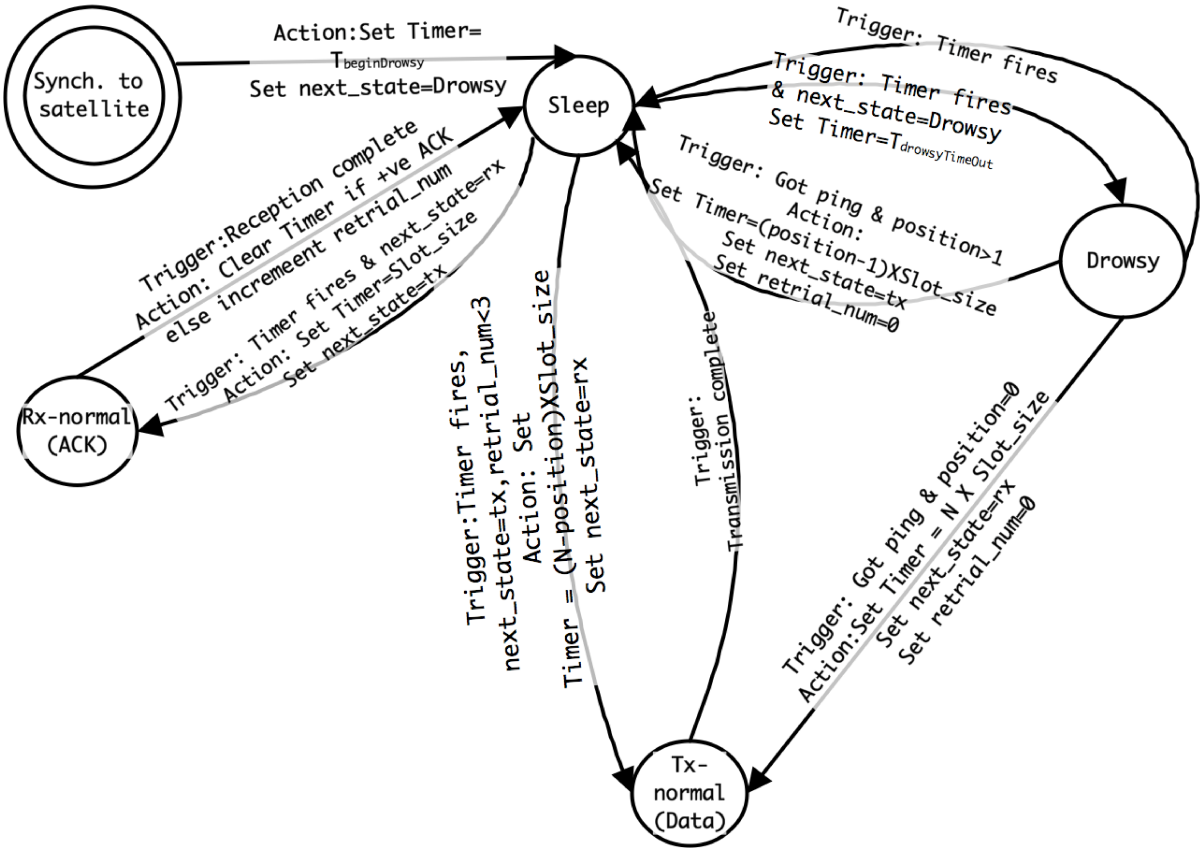
When the timer expires, the uplink node wakes up in drowsy mode while the downlink node wakes up in ping mode. Upon receiving the ping, the uplink node switches to the sleep mode and then to the normal transmit mode following a time-out of $SLOTSIZE$. After transmitting the ping, the downlink node switches to the normal receive mode. For the proof of concept, the values of $T_{schedule}$ and $SLOTSIZE$ are chosen so as to make the mode transitions conceivable by a human user.

After entering the normal transmit mode, the uplink node transmits a packet containing the sensing information, and at the same time displays the information on the LCD. The downlink node shows the received sensing information on the LCD, which can be verified to show the success of the transmission.

Upon the completion of transmission, the uplink and downlink nodes both go into the sleep mode, waiting for the next round of transmission. In [21], we presented the protocol design of our sensor network for automated collection of soil data from a farm-field.



(a) Finite state machine for receive protocol



(b) Finite state machine for send protocol

Figure 2.5 MAC protocol

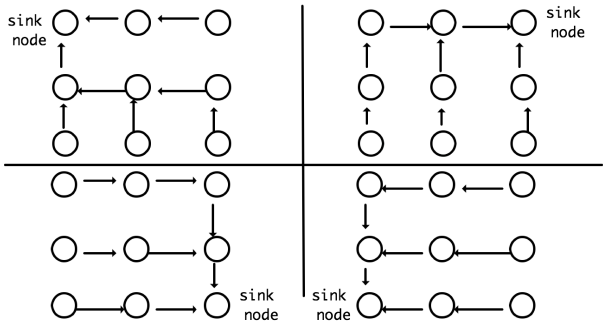


Figure 2.6 Possible routes over a macro-round comprising of 4 rounds.

Tx-type schedule	Scheduled time	Parent node id	Total number of senders to parent node	Position of node in transmitting to parent node	Slot size

Rx-type schedule	Scheduled time	Sender node ids (upto four)	Slot size

Figure 2.7 Routing/Schedule message structure.

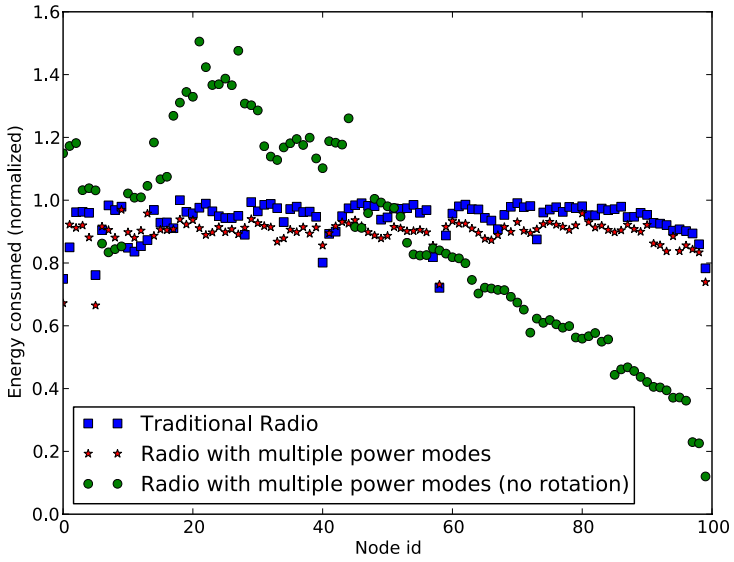


Figure 2.8 Energy consumption of 100 nodes over 10 macro-rounds.

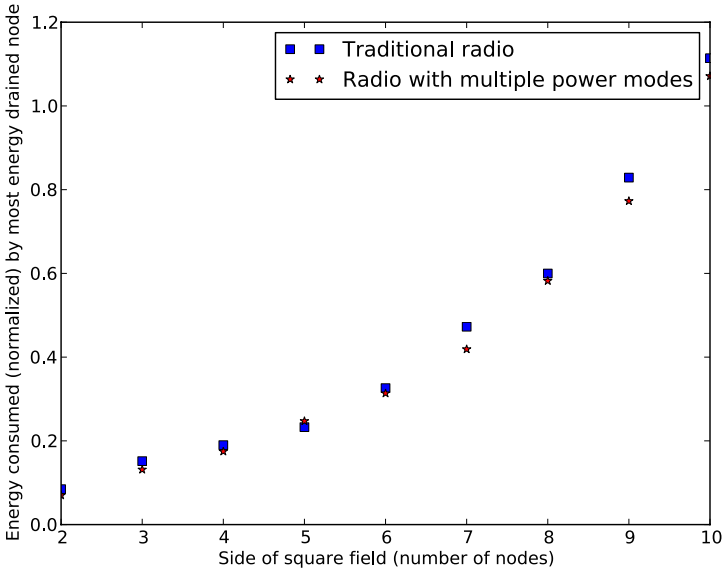


Figure 2.9 Energy consumed by most energy drained node versus field size.

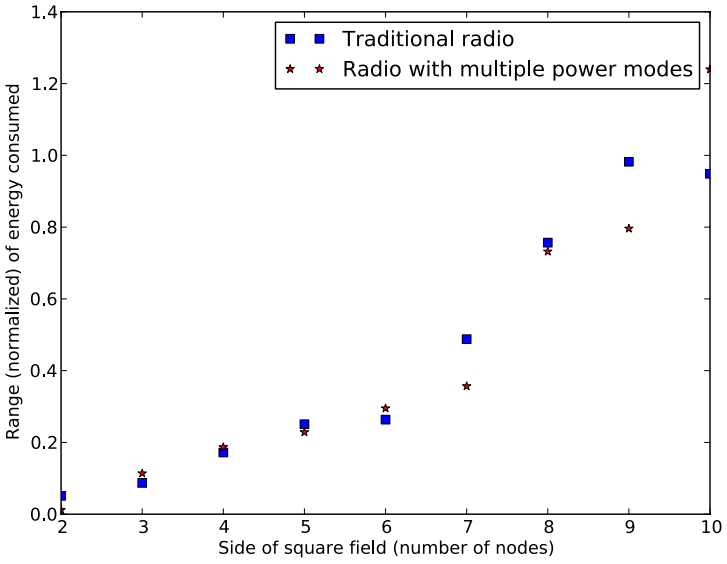


Figure 2.10 Difference between the energy consumed by most energy drained node and the least energy drained node versus field size.

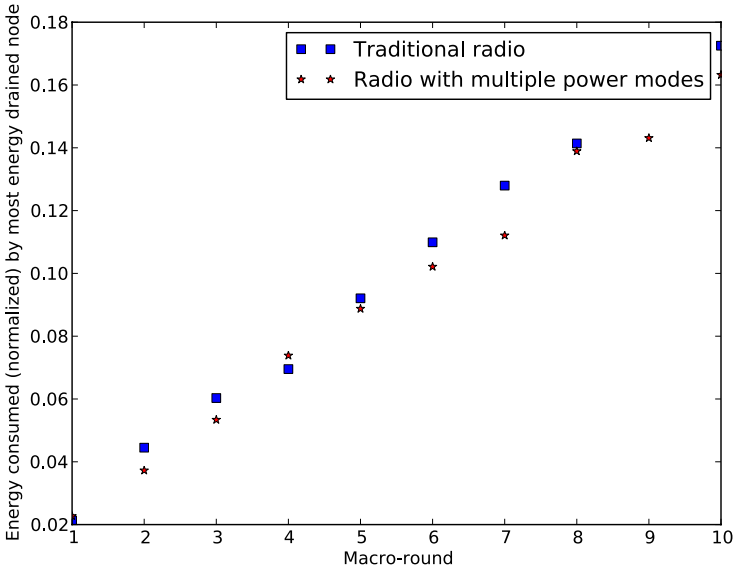


Figure 2.11 Energy consumed by most energy drained node versus number of macro-rounds.

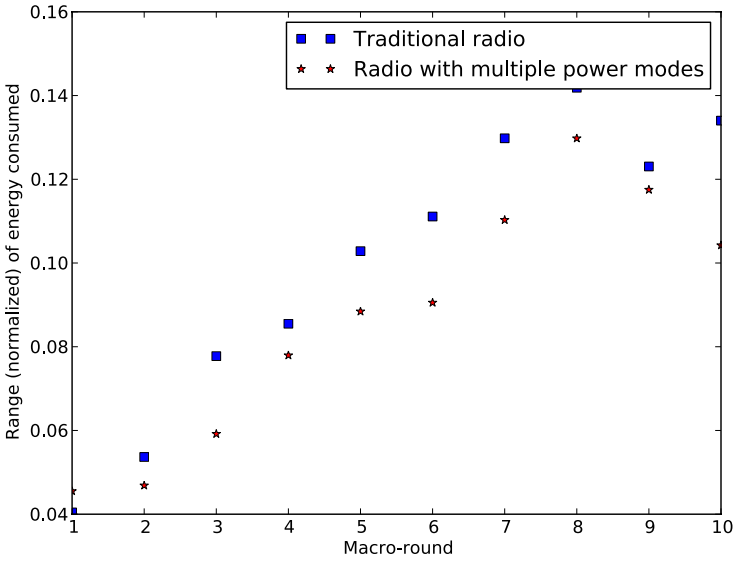


Figure 2.12 Difference between the energy consumed by most energy drained node and the least energy drained node versus number of macro-rounds.

(a) CC1110 evaluation module used in our implementation.

(b) SmartRF04EB evaluation board used in our implementation.

Figure 2.13 Sensor hardware.

CHAPTER 3. NETWORK PERFORMANCE MODELING AND EVALUATION

3.1 Introduction

Wireless transmission and reception are responsible for a major part of the energy consumption on a sensor node. The MAC layer is directly responsible for controlling access to the wireless channel and hence, is crucial for an energy efficient operation. In this chapter, we present an analytical approach to model the performance (throughput, latency and energy consumption) of our MAC protocol, and use the model to compare its performance with that of the well known S-MAC protocol. We also validate our mathematical models against simulation results using the event driven framework provided by TOSSIM [22].

The following are the contributions of this chapter:

- We present an analytical approach to model the performance (throughput, latency and energy consumption) of a MAC protocol under a given routing topology.
- We validate our analytical models by developing simulations in nesC and Python using the event driven framework provided by TOSSIM.

The results of our model and simulations show a 25% improvement in latency and a 65% improvement in energy consumption for the same level of throughput. Section 3.2 highlights related work in performance modeling and evaluation for wireless sensor network applications and Section 3.3 presents detailed analytical models for performance evaluation of PD-MAC and S-MAC. We derive models for throughput, delay and energy consumption for both protocols. The results of the analytical models and their validation using simulations are also presented in Section 3.3. The conclusions are summarized in Chapter 5.

3.2 Related work

Performance modeling of sensor networks has been addressed from various standpoints in existing works. For example, in [31], authors present a performance model to study the interaction between data aggregation and topology control by capturing tradeoffs among delay, energy, and fidelity of aggregation. In [32], authors investigate the system performance in terms of energy consumption, network capacity and data delivery delay. Both model the sensor behavior using Discrete Time Markov Chain models to capture the sleep/active dynamics and transmission and reception of packets. In [33], authors characterize the traffic load distribution of a sensor node as a function of its distance from the sink node in a network deployed over a line segment. In [34], authors present a Markov Chain model for a 802.11 wireless network to compute the throughput, delay and power consumption with focus on multicast services. In [35], authors present a framework to model the lifetime of sensor nodes as a function of time based on models for population dynamics in biological studies. In addition, several works exist where performance evaluation using simulation studies is performed. For example, see [36], [37], [38].

3.3 Mathematical analysis and simulation studies

We show the energy saved by PD-MAC by comparing its performance with that of a common duty-cycle based S-MAC protocol [1], in the setting of our application, using analytical models as well as simulations. For comparison purposes, both protocols operate under the same routing and scheduling information from the network layer. The network model used in this chapter is the same as the one presented in Section 2.3. The routing tree used in our performance evaluation studies is depicted in Figure 3.1. The nodes are placed in a grid pattern in the square field. Node 0 acts as the sink at which all the data from the network is collected. The scheduling strategy ensures that no two links in the network are active at the same time to avoid interference and also ensures that a node begins transmitting data to its downstream node only after it has already attempted to collect data from all of its upstream nodes. For

[1] Selected source code for this chapter's results can be found in Appendix A.

example, in Figure 3.1, link 17 → 13 is scheduled to be active after links 23 → 21 and 21 → 17, respectively.

The differences between S-MAC and PD-MAC lie in the implementation of the routes and schedules as well as the synchronization protocol. A synchronization protocol is used to synchronize two nodes that awaken to communicate with each other after a sleep period. We define the *communication time* as the time taken by a node to synchronize with and collect data from all its upstream senders. In S-MAC, the synchronization handshake and the data/ack exchange occurs in a pairwise manner between one sender node and one receiver node. If a node has more than one upstream neighbor, then the respective sender→receiver links are scheduled in a sequence. Thus, the total communication time for a given node is the sum of communication times corresponding to each of its upstream neighbors. On the other hand, in PD-MAC, a node synchronizes all of its upstream nodes with a common ping. See Section 2.4 to review the wake-up synchronization strategy employed in PD-MAC. A maximum number (N_s) of ping attempts are allowed in case of unsuccessful synchronization. Between successive pings, the receiver node allows the upstream neighbors a maximum of N_d data transmission attempts in a time division manner. This process of synchronization and data collection constitutes the communication time in PD-MAC. Additionally, as explained in Section 2.4, PD-MAC intelligently uses the multiple radio power modes to conserve energy.

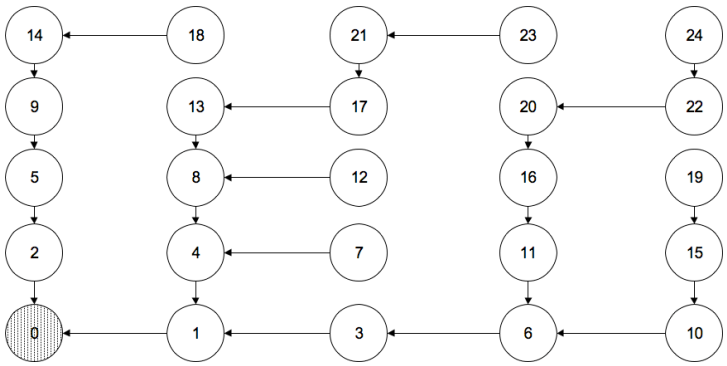


Figure 3.1 Routing tree R used for performance evaluation.

Figure 3.2 shows a possible execution scenario of a schedule consisting of one receiver and two sender nodes by S-MAC and PD-MAC. In S-MAC, the two senders are separately synchronized and then allowed a given number of attempts to transmit their data. When a

node wakes up to establish a communication link with its neighbor, it listens for a predefined duration before transmitting a synchronization request packet. The duration of this period is chosen long enough to accommodate the worst case drift (2Δ) between the clocks of the two nodes. That is, the first node that wakes up does not start transmitting until it is guaranteed that the other node on the link is awake to receive the synchronization request. Thus, $2\Delta + T_S$ suffices as the *discovery duration* (T_S denotes the duration of the synchronization request and reply packets), which is defined as the period of time a node must listen when it wakes up to discover any existing neighboring sleep-listen schedules (specific to S-MAC). The node that receives the synchronization request packet replies with a synchronization reply packet. Keeping the primary goal of comparing energy savings, we make a simplifying assumption that the reply packet is delivered error free. In addition, in regular S-MAC, the synchronization reply, which is the rebroadcast of the synchronization packet, by the node that receives a synchronization request happens after a random back-off to prevent multiple nodes from transmitting at the same time. In our application setting for S-MAC, only two nodes (i.e. sender and receiver) may be active at a time. Hence, a back-off is not needed. Once synchronization is established, the two nodes exchange data/ack packets for up to N_d attempts. In our comparison studies, for S-MAC, the nodes keep awake until they expend all synchronization and data transfer attempts or until data transmission succeeds. Since the data collection in the network is periodic, this setting ensures that S-MAC is not penalized with a longer round duration. A sender node is guaranteed to have data to send to its downstream neighbor when it wakes up eliminating the need for a sleep-listen schedule.

Figure 3.2 also shows a possible execution of the same scenario by PD-MAC. A common ping signal is sent by the receiver to synchronize the two senders. Sender 2 misses the first ping. After allowing $N_d = 1$ transmission attempt to both senders, when the receiver does not get all data, it transmits another ping signal ($N_s = 2$). Sender 2 captures this ping and successfully transmits its data this time, at which point the communication ends.

Now, we present our analytical performance models to compare the two MAC protocols.

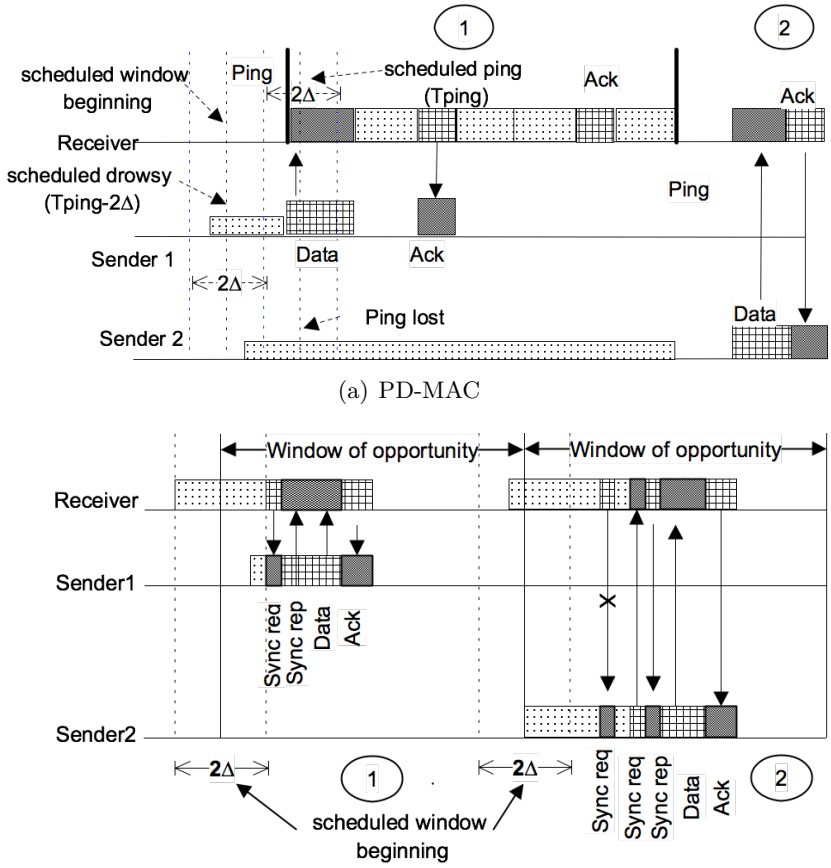


Figure 3.2 Execution scenario of multiple senders to one receiver communication links.

3.3.1 Modeling data-count at sink

The goal of the network and MAC protocols is to deliver all the data units generated in the sensor field to the sink node. However, due to channel noise and other sources of signal impairment some data units are lost. We model the number of data readings collected at each node in the network at the end of one round. $P_D^m(i)$ denotes the probability that node m stores i data readings (including the locally generated reading) just after it has attempted to collect data from its upstream neighbors, if any. In the remainder of the chapter, we use m to represent a downstream node and k for an upstream node. $S_m \subseteq U_m$ denotes the subset of all upstream neighbors U_m that are successful in sending data to node m. We evaluate $P_D^m(i)$ recursively as shown in Equation (3.1). The leaf nodes of the routing tree do not have any upstream neighbors. Therefore, the base case of the recursion is: $P_D^m(i) = 1$ for $i = 1$ and

$P_D^m(i) = 0$ for $i > 1$ for all leaf nodes m.

$$P_D^m(i) = \sum_{\substack{S_m \subseteq U_m: \\ \sum_{k \in S_m} i_k = i-1}} \left\{ \left(\prod_{k \in S_m} P_D^k(i_k) P_{\text{suc}}(i_k) \right) \left(\prod_{k \in U_m \setminus S_m} P_D^k(i_k) P_{\text{fail}}(i_k) \right) \right\} \tag{3.1}$$

The above expression computes the probability that a subset of upstream neighbors, $S_m \subseteq U_m$, successfully transmits $i-1$ data units to node m, such that the nodes in S_m collectively contain $i-1$ data units, for all such subsets S_m. $P_{\text{suc}}(i_k)$ denotes the probability that synchronization is established in at most N_s attempts and a data packet containing i_k data units is transmitted successfully in at most N_d attempts. $P_{suc}(i_k)$ can be calculated as:

$$P_{\text{suc}}(i_k) = \sum_{s=1}^{N_s} (1-q) q^{s-1} \sum_{r=1}^{N_d} \left((1-p_e)^{D(i_k)} (1 - (1-p_e)^{D(i_k)})^{r-1} \right), \tag{3.2}$$

where p_e is the bit error rate, $D(i_k)$ denotes the bit length of a data packet containing i_k data units. The probability of failure is given by: $P_{\text{fail}}(i_k) = 1 - P_{\text{suc}}(i_k)$. The expression for the data-count probability for PD-MAC and S-MAC remains the same except for $P_{\text{suc}}(i_k)$ and $P_{\text{fail}}(i_k)$. In Equation (3.2), the probability of failure of a synchronization attempt is denoted by q. For S-MAC, this is computed in accordance with the bit error probability, since the synchronization request packet contains encoded information about synchronization; while for PD-MAC, this denotes the probability of the failure of ping, which is a much lower value since a ping has the sole purpose of awakening the sender nodes (which will receive the ping) and it contains no other useful information to be decoded.

See Figure 3.3 for the expected data count at node 0 versus the number of synchronization attempts allowed for the two protocols. Both protocols have comparable performance as far as the data-count at the sink node is concerned. This behavior is expected because we allow the same number of synchronization and data transmission attempts for the two MAC protocols. The difference is in the value of the probability of synchronization. In S-MAC, the synchronization is established with a pair of packets transmitted at the regular power level while in PD-MAC a higher energy ping signal is used which has a lower probability of loss. Next, we compare the two protocols with respect to the delay.

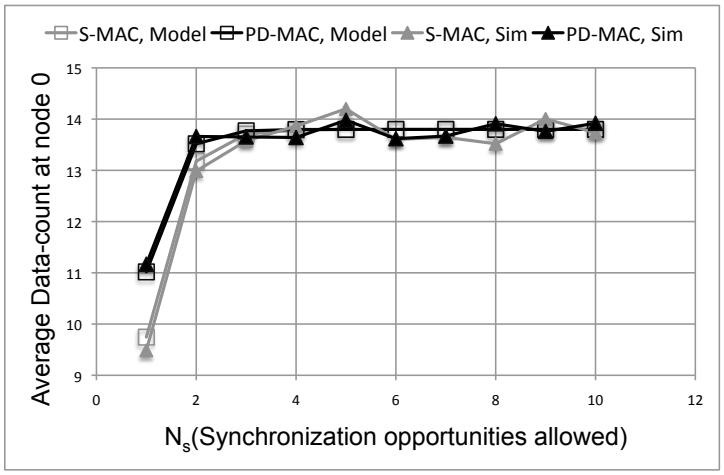


Figure 3.3 Expected data count stored at sink.

3.3.2 Modeling round duration

Round duration is the sum of the communication times for all nodes in the routing tree. In the following two subsections we model the communication time for a node for the two MAC protocols.

3.3.2.1 Modeling communication time for S-MAC

Recall that the total communication time for a node is given by the sum of the communication times corresponding to each of its upstream neighbors. Let D_{km} denote the communication time for a receiver node m, corresponding to an upstream neighbor k. We obtain a probability model for D_{km}. Let $P_{\text{sync}}(n)$ denote the probability that synchronization succeeds in the nth attempt ($n \leq N_s$) and $P_{\text{sync-fail}}$ denote the probability that synchronization fails all N_s attempts. If synchronization fails N_s attempts, the data phase does not take place. However, if synchronization succeeds in at most N_s attempts, the two nodes attempt to exchange the data packet for up to a maximum of N_d attempts. We model the expected communication time in Equation (3.3).

$$\mathbb{E}(D_{km}) = \sum_{i=1}^{N_s} P_{\text{sync}}(n) \times \left[T_{\text{sync}}(i) + T_{\text{data}}^{k} \right] + P_{\text{sync-fail}} \times T_{\text{no-sync}}, \tag{3.3}$$

where $T_{\text{sync}}(i)$ denotes the expected duration of the synchronization phase given that synchronization succeeds in the i^{th} attempt, T_{data}^{k} denotes the expected data phase duration

consisting of a maximum of N_d attempts, and $T_{\text{no-sync}}$ denotes the expected duration of the synchronization phase when synchronization fails all N_s attempts. In the latter case, no data phase takes place. Equation (3.4) captures $T_{\text{sync}}(i)$ which denotes the time period from the awakening of one of the nodes to synchronization in the i^{th} attempt.

$$T_{\text{sync}}(i) = \left\lfloor \frac{i+1}{2} \right\rfloor (2T_S + \Delta) + \{(i+1) \mod 2\} \mathbb{E}(Y) \tag{3.4}$$

Equation (3.4) can be understood as follows: the two nodes that are attempting to synchronize with each other take turns in sending synchronization request packets to each other. Therefore, if synchronization occurs in the i^{th} attempt, the node that starts first takes a total of $\lfloor \frac{i+1}{2} \rfloor$ turns, each of which lasts for the discovery duration of $2T_S + \Delta$. In addition, if the synchronization succeeds in an even-numbered attempt, a delay corresponding to the difference in the wake-up times of the two nodes, denoted Y in Equation (3.4), is also incurred. Similarly,

$$T_{\text{no-sync}} = \left\lfloor \frac{N_s+1}{2} \right\rfloor (2T_S + \Delta) + \{(N_s+1) \mod 2\} \mathbb{E}(Y). \tag{3.5}$$

Note that $Y = |X_k - X_m|$, where $X_k, X_m \sim U(0, 2\Delta)$ with pdf $f_{X_m}(x), f_{X_k}(x)$ are the wake-up times of nodes k and m, respectively. For example, Figure 3.4 shows $T_{\text{sync}}(5)$ and Y. We can derive the distribution of Y as follows:

$$\begin{aligned}
f_Y(y) &= \int_{0 \le x \le 2\Delta} [f_{X_m}(x-y) + f_{X_m}(x+y)] f_{X_k}(x) \mathrm{d}x \\
&= \int_{x=y}^{2\Delta} f_{X_m}(x-y) f_{X_k}(x) \mathrm{d}x + \int_{x=0}^{2\Delta-y} f_{X_m}(x+y) f_{X_k}(x) \mathrm{d}x \\
&= \int_{x=y}^{2\Delta} \frac{1}{2\Delta}\frac{1}{2\Delta} \mathrm{d}x + \int_{x=0}^{2\Delta-y} \frac{1}{2\Delta}\frac{1}{2\Delta} \mathrm{d}x \\
&= \frac{2\Delta - y}{2\Delta^2}
\end{aligned}$$

Then, the expected value of Y is given by:

$$\mathbb{E}(Y) = \int_0^{2\Delta} y \frac{2\Delta - y}{2\Delta^2} \mathrm{d}y = \frac{2}{3}\Delta.$$

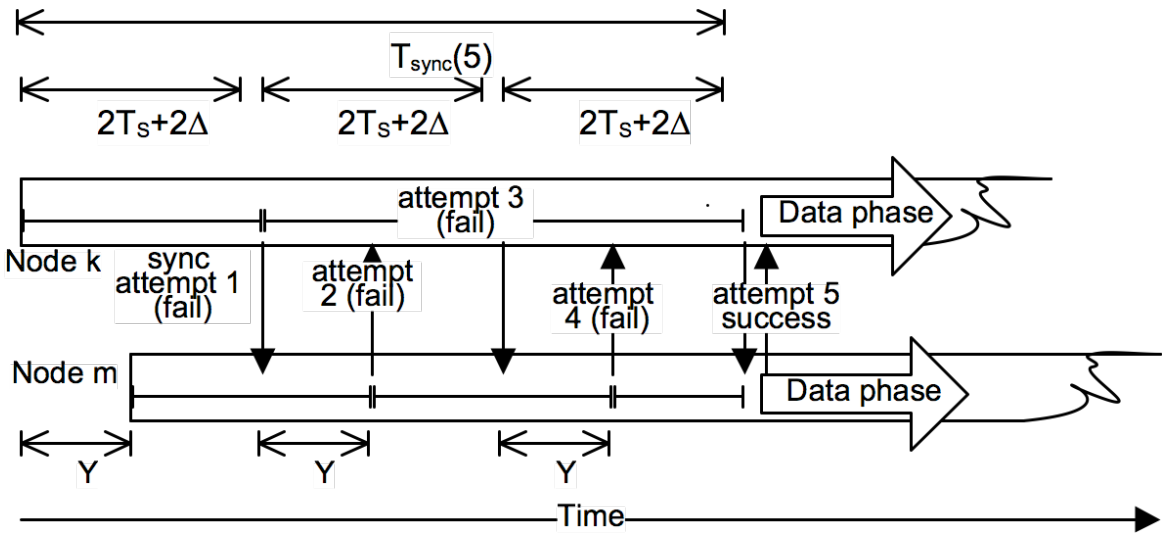


Figure 3.4 Synchronization phase in S-MAC. Y represents the absolute difference between the wake-up times of the sender and receiver nodes.

We now compute T^k_{data}. Let R_k denote the sub-routing-tree rooted at node k. Then, due to channel imperfections and features of the MAC protocol, node k may have any number of data packets $l \in \{1, 2, 3, ..., |R_k|\}$ when it is time to transmit to its downstream neighbor m. The probability distribution of the number of data packets contained at every node is expressed by Equation (3.1). The data slot duration for the link under consideration is denoted by $T_D^{|R_k|}$. The duration of the acknowledgement packet is denoted by T_A^1. Every data attempt takes a duration of $T_D^{|R_k|} + T_A^1$. Thus, we can model the expected data phase duration as follows:

$$T^k_{\text{data}} = \sum_{l=1}^{|R_k|} P_D^k(l) \left(\sum_{r=1}^{N_d} P^r_{\text{suc}}(l).r.(T_D^{|R_k|} + T_A^1) + P^{N_d}_{\text{fail}}(l).N_d.(T_D^{|R_k|} + T_A^1) \right), \tag{3.6}$$

where $P_D^k(l)$ is defined in Equation (3.1) and $P^r_{\text{suc}}(l)$ is the probability of successful transmission of a data packet containing l data units in the r^{th} attempt. Since the packet error rate for a data packet containing l data units is given by $p = 1 - (1 - p_e)^{D(l)}$, the formulae for $P^r_{\text{suc}}(l)$ and $P^r_{\text{fail}}(N_d)$ are as follows:

$$P^r_{\text{suc}}(l) = (1 - p)p^{r-1} \tag{3.7}$$

$$P^r_{\text{fail}}(N_d) = p^{N_d} \tag{3.8}$$

3.3.2.2 Modeling communication time for PD-MAC

Following the same line of approach as in the previous subsection, we model the communication time for a node. In PD-MAC, all the upstream neighbors $k \in U_m$ of node m attempt to synchronize with the common downstream neighbor at the same time and the nodes that are successfully synchronized transmit their data in a time division manner.

Let $T_m^i(S_m)$ denote the expected duration of time to synchronize the set of senders $S_m \subseteq U_m$ by node m and the associated data phases, given that $i-1$ synchronization attempts have already been made. The base case of the recursion is given by: $T_m^i(S_m) = 0$ for $i > N_s$ or $S_m = \phi$. $T_m^1(U_m)$ gives the communication time for node m. We have:

$$T_m^i(S_m) = \sum_{j=0}^{|S_m|-1} \binom{|S_m|}{j} (1-q)^j q^{|S_m|-j} \left[\sum_{\substack{S_j \subseteq S_m \\ :|S_j|=j}} \Big(N_d.T_{D+A}^m + P_{\text{suc}}^{S_j}.T_m^{i+1}(S_m - S_j) + (1 - P_{\text{suc}}^{S_j}).(N_s - i).N_d.T_{D+A}^m \Big) \right] + (1-q)^{|S_m|} D_m^1(S_m), \tag{3.9}$$

where $P_{\text{suc}}^{S_j}$ is the probability that all nodes from S_j succeed in their data transmission and T_{D+A}^m represents the duration of the entire data slot accounting for data transmissions from all senders and acknowledgement for the downstream node m. The first term in the above equation represents the case when at least one sender node is not synchronized in the current ping attempt. Of the senders that are synchronized, if all are successful in transmitting data in at most N_d attempts, an additional term $T_m^{i+1}(S_m - S_j)$ adds up to the communication time. However, if at least one node is unsuccessful in transmitting data in N_d attempts, $(N_s - i)N_d T_{D+A}^m$ adds to the communication time. The final term in the above equation represents the case when all the upstream neighbors of S_m are synchronized in the i^{th} ping. In that case, data transfer success may happen earlier (at $D_m^1(S_m)$) than when the next ping is scheduled to take place (at $(N_d.T_{D+A}^m)$). $D_m^r(S_m)$ denotes the time it takes to successfully deliver data from senders in the set $S_m \subseteq U_m$ that have successfully received the ping, given that $r-1$ data attempts have already been made. $D_m^1(S_m)$ is the portion of the communication

time in which senders that have just received the ping try to send their data and are allowed a maximum of N_d data transmission attempts interspersed with collective ACK/NACK packets. See Figure 3.2(a). In the following recursive model, $D_m^r(S_m) = 0$ if $r > N_d$ or $S_m = \phi$ gives the base case.

$$D_m^r(S_m) = \sum_{j=0}^{|S_m|} \left(\sum_{\substack{\forall S_j \subset S_m \\ |S_j|=j}} \left[\sum_{k \in S_j} \sum_{l=1}^{|R_k|} P_D^k(l)(1-p_e)^{D(l)} \sum_{k \in S_m \setminus S_j} \sum_{l=1}^{|R_k|} P_D^k(l)(1-(1-p_e)^{D(l)}) \right] \times \right.$$
$$\left. (T_{D+A}^m + D_m^{r+1}(S_m - S_j)) \right) \tag{3.10}$$

The above equation models the portion of communication time for receiver node m just after the point when the last set of un-synchronized senders S_m has just received the ping. From the set S_m of upstream neighbors, any number j of nodes may be successful in the current data attempt while the remaining fail.

The results of the model and simulation for both S-MAC and PD-MAC are shown in Figure 3.5. The figure compares the round duration for the two protocols versus the number of synchronization attempts allowed. It should be noted that PD-MAC achieves a 25% shorter round duration than S-MAC.

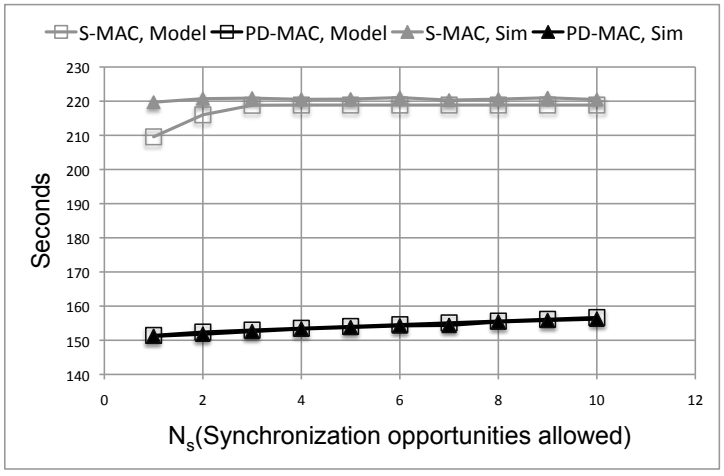


Figure 3.5 Expected round duration vs. number of synchronization opportunities.

3.3.3 Computation of energy consumption

To model the energy consumption of a node, we model the duration of time spent in various radio modes; idle, transmit, receive, drowsy and ping. Knowing the current drawn in various radio modes and assuming that the voltage is a constant, the product of the current drawn and the time spent in a particular mode gives an estimate of the energy consumed. We now model the time spent in different radio modes for the two protocols, separately.

3.3.3.1 Energy consumption model for S-MAC

The operation of S-MAC can be divided into synchronization and data phases for the purposes of energy consumption modeling. In the following two subsections we model the time spent by a node in all possible radio modes in the two phases.

Synchronization phase. An example of a synchronization phase in which synchronization succeeds in the fifth attempt is shown in Figure 3.4. In the synchronization phase, a node spends time in three radio modes; idle listen, transmit mode (sync packets) and receive mode (sync packets). The expected duration of idle mode and expected number of synchronization request/reply packets sent/received are obtained in a very similar fashion as in Equations (3.4) and (3.5) and we leave them out to avoid repetition.

Data phase. Once synchronization is established, the two nodes enter the data phase. In this phase, a node is either in transmit mode or receive mode. First, we model the expected duration spent by the upstream node k transmitting the data packet. This is also the expected duration spent by a node m receiving a data packet from its upstream neighbor k. We have:

$$T^k_{\text{tx-data}} = \sum_{l=1}^{|R_k|} P^k_D(l) \left\{ \sum_{r=1}^{N_d} P^r_{\text{suc}}(l).r.T_D(l) + P^{N_d}_{\text{fail}}(l).N_d.T_D(l) \right\} \tag{3.11}$$

$$= T^m_{\text{rx-data}}, \tag{3.12}$$

where $T_D(l)$ denotes the duration of a data packet containing l data units and $P^r_{\text{suc}}(l)$ and $P^{N_d}_{\text{fail}}(l)$ are defined in Equation (3.8). The above equation models the data transmit duration by considering all possible values of the data units stored at the sender node and the number of data attempts made. Similarly, we can model the expected duration of time spent by the

downstream node sending an acknowledgement packet. This is also the expected duration spent by the upstream node receiving the acknowledgement packet:

$$T^m_{\text{tx-ack}} = \sum_{l=1}^{|R_k|} P^k_D(l) \left(\sum_{r=1}^{N_d} P^r_{\text{suc}}(l).r.T^1_A + P_{\text{fail}}(l).N_d.T^1_A \right)$$
$$= T^k_{\text{rx-ack}}. \tag{3.13}$$

3.3.3.2 Energy consumption model for PD-MAC

Each node in PD-MAC spends energy while operating as a downstream node or as an upstream node. We model the energy consumption for the two cases separately.

Energy consumption for upstream node. An upstream node consumes energy in three modes; drowsy, data transmission and acknowledgement reception.

Given that synchronization is successful, expected duration of node k's data transmit mode can be modeled exactly as in Equation (3.11). Next, we model the expected duration of time $T^k_{\text{rx-ack}}$ spent in receive mode by an upstream neighbor k (given that synchronization is successful):

$$T^k_{\text{rx-ack}} = \sum_{l=1}^{|R_k|} P^k_D(l) \left(\sum_{r=1}^{N_d} r T_A^{|U_m|} P^r_{\text{suc}}(l) + N_d T_A^{|U_m|} P^{N_d}_{\text{fail}}(l) \right),$$

where $T_A^{|U_m|}$ is the duration of the acknowledgement packet transmitted by node m. The expected duration of time spent in drowsy mode is modeled by the following equation:

$$T^k_{\text{drowsy}} = 2\Delta(1-q) + \sum_{i=2}^{N_s} q^{i-1}(1-q).i.N_d.T^m_{D+A}, \tag{3.14}$$

where T^m_{D+A} is same as defined for Equation (3.9). The first term in the above equation accounts for the case when the synchronization happens in the first ping attempt and the summation represents the case when the synchronization happens in the i^{th} attempt ($2 \leq i \leq N_s$). When a sender misses a ping from its downstream neighbor, it has to wait N_d times the duration of the entire data slot for the next ping transmission.

Energy consumption for downstream node. A downstream node consumes energy in four modes; ping, idle listen, data receive and acknowledgement transmit modes. We model the duration of all these modes.

First, we model the number of pings transmitted by downstream node m. Downstream node attempts to synchronize its upstream nodes until it receives data from all for up to a maximum of N_s attempts. Let $S_m \subseteq U_m$ represent the set of upstream neighbors that have not yet received the ping from node m and $i-1$ be the number of pings already transmitted by node m. We recursively model the expected number of pings transmitted by node m. $N^m_{\text{ping}}(S_m, i) = 0$ for $S_m = \phi$ or $i > N_s$. $N^m_{ping}(U_m, 1)$ gives the expected number of pings transmitted by node m. We have:

$$N^m_{\text{ping}}(S_m, i) = \sum_{j=0}^{|S_m|} (1-q)^j q^{|S_m|-j} \sum_{\substack{S_j \subset S_m \\ :|S_j|=j}} \left\{ \sum_{k \in S_j} \left[\sum_{l=1}^{|R_k|} P^k_D(l) \right. \right.$$
$$\left. P^{\leq N_d}_{\text{suc}}(l) \{1 + N^m_{\text{ping}}(S_m - S_j, i+1)\} \right] +$$
$$\left. \left[1 - \sum_{k \in S_j} \sum_{l=1}^{|R_k|} P^k_D(l) P^{\leq N_d}_{\text{suc}}(l) \right] (N-i) \right\},$$

where $P^{\leq N_d}_{\text{suc}}(l)$ is the probability of successful transmission of l data units. S_j is the set of nodes that successfully receive the ping in the current ping attempt. If at least one of the nodes in S_j is unsuccessful in transmitting data in N_d attempts, then node m transmits $(N-i)$ additional pings since loss of data is treated as synchronization loss at node m. Otherwise, the recursive expression is used. The expected duration of the data receive mode for node m is the same as the sum of the expected durations of the data transmit modes for all the upstream neighbors of node m:

$$T^m_{\text{rx-data}} = \sum_{k \in U_m} T^k_{\text{tx-data}}, \tag{3.15}$$

where $T^k_{\text{tx-data}}$ is modeled by Equation (3.11).

Now, we model the number of acknowledgement packets transmitted by node m. $N^m_{\text{tx-ack}}(S_m, i)$ represents the number of acknowledgement packets transmitted by node m, given that the subset S_m of the upstream neighbors is unsuccessful in receiving the ping so far and $i-1$ pings have been transmitted by node m. The total expected number of ping packets transmitted by

node m is then given by $N^m_{\text{tx-ack}}(U_m, 1)$. We have:

$$N^m_{\text{tx-ack}}(S_m, i) = P_{\text{ping-suc}}(S_m).\left[\sum_{r=1}^{N_d} P^r_{\text{data-suc}}(S_m).r + (P^{N_d}_{\text{data-fail}}(S_m)).(N-i)N_d\right] + \sum_{j=0}^{|S_m|-1} \sum_{\substack{\forall S_j \subseteq S_m \\ :|S_j|=j}} P_{\text{ping-suc}}(S_j)\Big((P^{N_d}_{\text{data-fail}}(S_j)).(N-i).N_d + P^{\leq N_d}_{\text{data-suc}}(S_j).(N_d + N^m_{\text{tx-ack}}(S_m - S_j, i+1))\Big), \quad (3.16)$$

where $P_{\text{ping-suc}}(S_j)$ denotes the probability that the subset S_j of senders receives the ping during the current ping attempt, $P^r_{\text{data-suc}}(S_j)$ denotes the probability that the set of senders S_j succeeds in data transmission such that the last successful sender succeeds in the r^{th} attempt and $P^{N_d}_{\text{data-fail}}(S_j)$ denotes the probability that at least one of the nodes in the set S_j fails all the N_d data transmission attempts. The above equation models the fact that node m transmits acknowledgement packets until it receives data from all of its upstream neighbors. If any of the nodes from the set of upstream neighbors misses the ping, node m transmits $N_d(N_s - i)$ additional acknowledgements (making $N_d N_s$ acknowledgements in total) and uses up all N_s ping attempts. If set S_m is the last subset of upstream neighbors to be synchronized, then the upstream neighbor that succeeds last in its data transmission determines the number of acknowledgements transmitted by node m.

Note that the expected idle duration of node m is simply given by the difference of the total awake time and the sum of the expected transmit and receive durations of node m. Node m stays awake until it receives data from all of its upstream neighbors or the number of synchronization attempts expire. In other words, node m stays awake until it transmits the last acknowledgement packet. Hence, an expression for the total awake time can be obtained using a minor modification of Equation (3.16) and we omit it to avoid repetition.

The results of the energy consumption model are shown in Figure 3.6. Energy savings of approximately 65% are predicted for PD-MAC.

To compute the energy consumption, we used the values for the current drawn from the battery as per the datasheet of CC1110 for various modes of the radio [23]. A bit error rate of 0.01, ping transmission error probability of 0.1, 8-bits per data unit, 8-header bits per

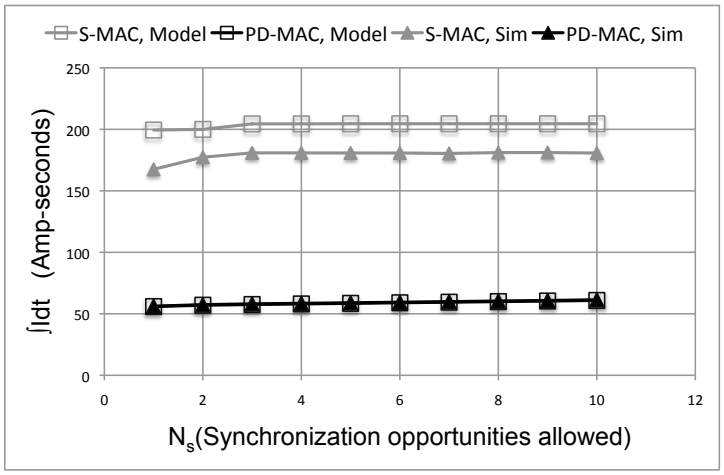


Figure 3.6 Expected energy consumption per node.

packet, ping pulse duration of 0.1 s, and a baud-rate of 1200 bps was used for our performance evaluation studies. For the proof of concept, we have implemented PD-MAC for the case of two nodes, i.e., one uplink node and one downlink node, see [21] for more details.

CHAPTER 4. NETWORK BASED SENSOR NODE LOCALIZATION

4.1 Motivation and related work

Self localization capability of sensor nodes is a highly desirable characteristic. Location information is an integral part of the sensor data collected from spatially distributed points in a large field, and also useful for the correct functioning and improved performance of the network stack. For instance, our routing strategy chooses the next hop node from neighboring nodes based on their geographical proximity to the sink. Recent publications reveal several examples motivating the need for localization as a key element to enhance the functionality of different layers of the sensor network system. For example, location based routing techniques proposed for wireless sensor networks require location data as an integral input, as seen in [39], [40], [41], [42]. Several security schemes also rely on location awareness of the sensor nodes. In [43], authors propose an end-to-end data security mechanism which involves binding the secret keys stored at a node to its location.

In [44], authors study sensor node localization from a statistical signal processing standpoint. They present a description of the measurement based statistical models to describe the three common measurement technologies used for ranging in sensor node localization: time of arrival (ToA), angle of arrival (AoA) and received signal strength (RSS). The models are used to derive Cramer-Rao lower bounds on the location estimates' variances. In [45], authors address node localization problems from a theoretical foundational standpoint. They derive the conditions for unique localizability of a network and also study the computational complexity of network localization.

Localization approaches can be broadly categorized as anchor free or anchor based. Anchor free approaches do not rely on any node in the network with prior location information. Several

anchor free localization schemes have been proposed in literature [46], [47], where nodes estimate their locations in a relative coordinate system formed by a few chosen nodes in the system. In [47], authors formulate localization as local optimization problems that are solved at the cluster heads for building the local coordinate systems of nodes belonging to their respective clusters. Nodes estimate distances to their neighbors using time of arrival measurements. As a second stage, cluster heads collaborate to build a global map of the network by combining the local maps obtained in the first stage.

Anchor based node localization, on the other hand, relies on a number of anchor nodes in the network that are assumed to have prior location information [48], [49], [50], [51]. In [52], authors present an anchor based range free localization scheme for a sensor network in an environment with obstacles. Range free localization techniques do not use ranging measurements such as time of arrival, signal strength and angle of arrival. For example, hop count information can be used for approximating the distance between two nodes once the hop length is estimated using anchor nodes. Alternatively, centroid based schemes can be used to estimate location coordinates given a set of anchor nodes in proximity. In [52], authors argue that for a sensor network deployment in an environment with obstacles leading to a concave network shape, the shortest packet delivery path distance may not be approximated by the geographical Euclidean distance. They propose a scheme to identify line of sight anchors from non-line of sight anchors in order to ignore the hop count measurements from the latter to estimate locations. In [53], authors propose an anchor based scheme based on multidimensional scaling (MDS) [54] that uses connectivity information of the neighboring nodes to localize a network of n nodes with a time complexity of $O(n^3)$. A distributed variant is also presented that builds local maps of nodes and then patches them together to form a global map. Another advantage of the distributed approach is that smaller local maps are more accurate for irregular networks because connectivity information for far away nodes is ignored. Least squares optimization is incorporated to further improve the accuracy at the expense of additional computation.

Range based localization techniques require special hardware on each node to measure a ranging signal and estimate the distance to the transmitting node. The advantage is increased localization accuracy compared to range free localization. An example is a controlled event

driven localization system where detectable events are generated using an event disseminator. Nodes report the detection of the events along with the timestamps to a central location server which computes the position of the nodes based on its own knowledge of the event propagation in the sensor field. In [55], authors present the implementation results of a localization system based on the detection of light generated by such a controlled event distribution system. The detection time at a sensor node is used to arrive at a spatial relationship between the node and the event generator. In [56], authors present the design of a localization system using uncontrolled events, which is more suitable for an outdoor environment where controlled dissemination of events may not be readily feasible. In this method, a small number of anchor nodes in the field are first used to estimate the generation parameters of the uncontrolled events. Thereafter, controlled event localization techniques can be used to estimate the nodes' unknown location coordinates.

In [57], authors present an application of cooperative localization techniques to UWB (ultra wide bandwidth) wireless networks. The authors quantify the performance of several algorithms based on the ranging models available for UWB technology. They also present a localization algorithm by mapping a statistical model for graphical inference onto the network topology. In [58], authors present a sequential Monte Carlo localization method for sensor networks with mobile nodes that uses both range measurements and hop distance. The filtering distribution of the nodes' location coordinates is determined given range based and range free information along with mobility information.

Different applications may have varying requirements for the degree of localization accuracy desired. A trade-off may need to be made between the complexity of the localization hardware, algorithm complexity, deployment costs and localization accuracy. In [59], authors present a study of the error inducing parameters in sensor node localization. The authors derive the Cramer-Rao bound for Gaussian measurement error in multi-hop localization systems using distance and angular measurements. They study the effect of the parameters such as measurement technology accuracy, node density, beacon node concentration, and beacon uncertainty on the localization error. See [60] for a survey of the latest research research in the area of sensor node localization.

To the best of our knowledge, none of the existing work in literature has approached sensor node localization for a network where the sensor nodes are placed in different physical media. In this chapter, we present our approach to sensor node localization for such a network in which the sensor nodes are buried underground while the anchor nodes are located above ground.

The following are the contributions of this chapter:

- We present statistical 3-D localization frameworks based on received signal strength (RSS) and time of arrival (ToA) measurements for multiple media (air and soil) and multiple reflected paths in a lossy medium (soil).
- For the RSS framework, we develop the Rician fading models for multi-media (air to soil) and multi-path (soil to soil) communication.
- For the ToA framework, we extend the maximum likelihood estimation framework, that involves cross-correlation of transmitted versus received signal, to the multi-path case.
- We present simulation results that implement the proposed localization schemes in Python, including error analysis.
- We present sensitivity analyses of the estimated location coordinates with respect to various parameters of interest.

Section 4.2 presents our localization framework based on received signal strength measurements. Section 4.3 presents our localization framework based on time of arrival measurements. Section 4.4 presents the simulation results, including sensitivity analysis in Section 4.4.1[1]. Conclusions are presented in Chapter 5.

4.2 Localization based on received signal strength

4.2.1 Notation and preliminaries

A wireless signal experiences fading that is a combination of large-scale and small-scale fading [61]. The former is due to large-scale uncertainties such as obstacles, weather etc.

[1]Selected source code for this chapter's results can be found in Appendix B.

whereas the latter is due to small-scale uncertainties due to multi-path interference. The large-scale power fading is known to cause the received power to be lognormally distributed, whereas the small-scale power faded received signal power follows a non-central χ^2 distribution with 2 degrees of freedom (equivalently, the amplitude has a Rician distribution).

Accordingly, the signal power, R_{mn}, received at node n and transmitted from node m, has the following combined distribution:

$$f_{R_{mn}}(r_{mn}) = \int_0^\infty f^S_{R_{mn}}(r_{mn}; p_{mn}, \kappa_{mn}) f^L_{P_{mn}}(p_{mn}; \mu_{mn}, \sigma_{mn}) dp_{mn}, \tag{4.1}$$

where $f^S_{R_{mn}}(r_{mn}; p_{mn}, \kappa_{mn})$ is the non-central χ^2 probability density function (pdf) modeling the effect of small scale fading with mean received power p_{mn} from the specular (line-of-sight) and scatter (non-line-of-sight) components, and κ_{mn} is the specular power to scatter power ratio; and $f^L_{P_{mn}}(p_{mn}; \mu_{mn}, \sigma_{mn})$ is the lognormal probability distribution function modeling the effect of large scale fading with parameters μ_{mn} and σ_{mn}. The distributions are expressed as follows, where the notation $I_0(\cdot)$ denotes the modified Bessel function of the first kind and order 0:

$$f^S_{R_{mn}}(r_{mn}; p_{mn}, \kappa_{mn}) = \frac{1+\kappa_{mn}}{p_{mn}} e^{-\left(\frac{(\kappa_{mn}+1)r_{mn}}{p_{mn}} + \kappa_{mn}\right)} I_0\left(\sqrt{\frac{4(\kappa_{mn}+1)\kappa_{mn} r_{mn}}{p_{mn}}}\right) \tag{4.2}$$

$$f^L_{P_{mn}}(p_{mn}; \mu_{mn}, \sigma_{mn}) = \frac{1}{p_{mn}\sqrt{2\pi\sigma^2_{mn}}} e^{-\frac{(\ln p_{mn} - \mu_{mn})^2}{2\sigma^2_{mn}}}. \tag{4.3}$$

Lognormal fading occurs over slow time-scales while multi path fading occurs over much shorter time scales [61]. In a short time frame, say within the duration of a round when the node pairs communicate to gather the information about path losses from the received signal powers (for example, to enable position estimation), the large-scale fading effect can be ignored by treating its distribution to have zero-variance ($\sigma_{mn} = 0$; so the distribution $f^L_{P_{mn}}(p_{mn}; \mu_{mn}, 0)$ becomes an impulse centered at the mean value, $p_{mn} = e^{\mu_{mn}}$). In this case, the received power's pdf simplifies to:

$$f_{R_{mn}}(r_{mn}) = f^S_{R_{mn}}(r_{mn}; p_{mn}, \kappa_{mn}), \tag{4.4}$$

which is the fading model that we rely on for the short time periods when the localization measurements are taken.

In general, the received signal has in-phase as well as quadrature phase envelopes X_{in} and X_{quad} that are both Gaussian random variables. This follows from the application of the Central Limit Theorem, as the received signal is the superposition of a large number of scattered components traveling along a multitude of paths [62]. Thus, the overall signal envelope $X = X_{in} + jX_{quad}$ is complex gaussian. If the signal travels over a direct and a reflected path, the received signal envelope will be the complex gaussian $X + X^{(r)}$, where $X^{(r)} = X^{(r)}_{in} + jX^{(r)}_{quad}$ corresponds to the reflected path. Then, the overall average power, p_{mn}, of the received signal is given by:

$$\begin{aligned}
\mathbb{E}[|X + X^{(r)}|^2] &= \mathbb{E}[(X + X^{(r)})(X + X^{(r)})^*] \\
&= \mathbb{E}[XX^*] + \mathbb{E}[X^{(r)}X^{(r)*}] + \mathbb{E}[XX^{(r)*}] + \mathbb{E}[X^{(r)}X^*] && (4.5)\\
&= P_X + P_{X^{(r)}} + \mathbb{E}[X]\mathbb{E}[X^{(r)*}] + \mathbb{E}[X^{(r)}]\mathbb{E}[X^*] && (4.6)\\
&= P_X + P_{X^{(r)}} + V_X e^{j\phi_X} V_{X^{(r)}} e^{-j\phi_{X^{(r)}}} + V_X e^{-j\phi_X} V_{X^{(r)}} e^{j\phi_{X^{(r)}}} \\
&= P_X + P_{X^{(r)}} + V_X V_{X^{(r)}} e^{j(\phi_X - \phi_{X^{(r)}})} + V_X V_{X^{(r)}} e^{-j(\phi_X - \phi_{X^{(r)}})} \\
&= P_X + P_{X^{(r)}} + 2V_X V_{X^{(r)}} \cos(\phi_X - \phi_{X^{(r)}}) && (4.7)\\
&= P_X + P_{X^{(r)}} + 2\sqrt{P_X P_{X^{(r)}} \frac{\kappa}{\kappa+1}\frac{\kappa^{(r)}}{\kappa^{(r)}+1}} \cos(\phi_X - \phi_{X^{(r)}}), && (4.8)
\end{aligned}$$

where (4.5) - (4.6) follows from assuming that X and $X^{(r)}$ are independent; $V_X e^{j\phi_X}$ and $V_{X^{(r)}} e^{j\phi_{X^{(r)}}}$ represent the specular (line-of-sight) components of X and $X^{(r)}$, respectively, in polar form. (4.8) follows from the following: the specular component power, V_X^2 is related to the total average power P_X by the Rician factor as: $V_X^2 = \frac{\kappa}{\kappa+1} P_X$. Similarly, $V^2_{X^{(r)}} = \frac{\kappa^{(r)}}{\kappa^{(r)}+1} P_{X^{(r)}}$.

Note that while the received signal envelope $X + X^{(r)}$ is complex Gaussian, its amplitude has a Rician distribution and its power has a non-central χ^2 distribution with 2 degrees of freedom. In the special case that there is no line of sight path (so the received signal has no "specular" component and only the "scatter" component), the amplitude becomes Rayleigh distributed (as a special case of Rician) and the power becomes exponentially distributed (as a special case of non-central χ^2), given as follows:

$$f_{R_{mn}}(r_{mn}) = f^S_{R_{mn}}(r_{mn}; p_{mn}, 0) = \frac{1}{p_{mn}} e^{-p_{mn} r_{mn}}. \tag{4.9}$$

In the following sections we express the parameters of the probability distribution of the received signal power in terms of the location coordinates of the sensor nodes in order to formulate the estimation problem for quantities of interest. The received power pdf given by Equation (4.4) has two parameters; the average received power p_{mn} and the Rician K-factor κ_{mn}. Existing works study the variation of the Rician K-factor with distance [63], [64] and its estimation [65], [66], [67], [68]. However, we assume it to be a known constant for the purposes of localization in our application. In the following section we express p_{mn} in terms of the location coordinates of the sender and receiver nodes.

4.2.2 Mean received power in terms of locations

It is known that the mean signal power $p_{mn} = e^{\mu_{mn}}$ decays with distance along a single path and within a single lossless medium following the power law [69]:

$$p_{mn}(d) = \eta d^{-k} \tag{4.10}$$

where d is the distance between the two nodes m and n, η is a constant and k is the path loss exponent, both of which are functions of the medium. For example, in free space:

$$k = 2 \text{ and } \eta = \frac{p_m G_m G_n \lambda^2}{(4\pi)^2}, \tag{4.11}$$

where p_m is the sender power, G_m and G_n are the sender and receiver antenna gains, and λ is the wavelength. For simulation purposes we use, $\eta = (p_m\lambda^2)/(4\pi)^2$ (ie, $G_m = G_n = 1$) and $k = 2$. A lossy dielectric medium such as soil, has additional attenuation due to conductivity losses [69]. The wave propagation equation for such a medium is given by:

$$E(r,t) = E_0 e^{-\alpha r}\cos(\omega t - \beta r), \tag{4.12}$$

where E is the electric field at time t at a distance r from the source which transmits at amplitude E_0. The complex propagation constant of the lossy dielectric medium, such as soil, is $\alpha + j\beta$, where:

$$\alpha = \omega\sqrt{\frac{\mu\epsilon'}{2}\left[\sqrt{1+\left(\frac{\epsilon''}{\epsilon'}\right)^2}-1\right]}, \tag{4.13}$$

and

$$\beta = \omega \sqrt{\frac{\mu\epsilon'}{2}\left[\sqrt{1+\left(\frac{\epsilon''}{\epsilon'}\right)^2}+1\right]}, \tag{4.14}$$

where μ is the permeability, and ϵ' and ϵ'' are the real and imaginary parts of the complex permittivity of the medium of propagation.

Hence, in the more general setting where $\alpha \neq 0$, Equation (4.10) takes the form:

$$p_{mn}(d) = \eta d^{-k} e^{-2\alpha d}. \tag{4.15}$$

In the following we extend this calculation to the case of multiple paths as well as multiple media. In the notation below, we introduce the superscripts a and s, respectively, when referring to the air and the soil as the medium.

4.2.2.1 Multi-path extension: soil to soil communication

When the sender node m as well as the receiver node n are within soil as shown in Figure 4.1, there exist the direct and the reflected paths and consequently following Equation (4.8), the average received power is given by:

$$p_{mn}^{(ss)} = p_{mn}(d_{mn}) + p_{mn}(d_{mn}^{(r)})\rho + 2\sqrt{p_{mn}(d_{mn})p_{mn}(d_{mn}^{(r)})\rho\frac{\kappa_{mn}}{\kappa_{mn}+1}\frac{\kappa_{mn}^{(r)}}{\kappa_{mn}^{(r)}+1}}\cos\frac{2\pi}{\lambda^{(s)}}(d_{mn}^{(r)} - d_{mn}), \tag{4.16}$$

where

$$\rho = \left(\frac{\sqrt{\mu_a/\epsilon'_a} - \sqrt{\mu_s/\epsilon'_s}}{\sqrt{\mu_a/\epsilon'_a} + \sqrt{\mu_s/\epsilon'_s}}\right)^2, \tag{4.17}$$

is the reflection coefficient, and d_{mn} and $d_{mn}^{(r)}$ are the distances of the direct and reflected paths respectively (see Figure 4.1) which are given by:

$$d_{mn} = \sqrt{(x_m - x_n)^2 + (y_m - y_n)^2 + (z_m - z_n)^2}, \tag{4.18}$$

and

$$d_{mn}^{(r)} = \sqrt{[(x_m - x_n)^2 + (y_m - y_n)^2]\frac{z_m^2}{(z_m+z_n)^2} + z_m^2} + \sqrt{[(x_m - x_n)^2 + (y_m - y_n)^2]\frac{z_n^2}{(z_m+z_n)^2} + z_n^2}. \tag{4.19}$$

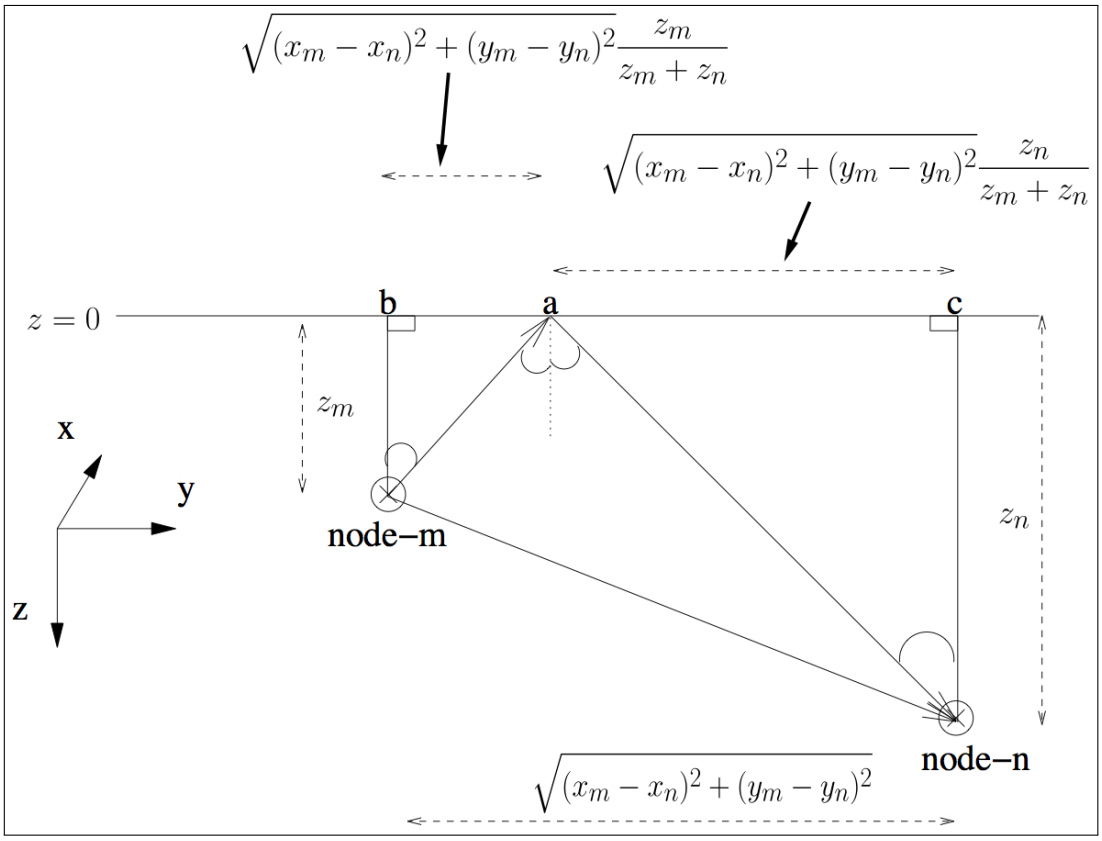


Figure 4.1 Soil to soil communication between nodes m and n: Direct vs. reflected paths

Also, the parameter $\kappa_{mn}^{(ss)}$ measuring the ratio of the specular to scatter power in the received signal can be derived by noting that the total average received power is $p_{mn}^{(ss)}$ as calculated in Equation (4.16), while the total received scatter power is simply the sum of the scatter powers for the two paths: $\frac{p_{mn}^{(s)}(d_{mn})}{\kappa_{mn}+1} + \frac{p_{mn}^{(s)}(d_{mn}^{(r)})}{\kappa_{mn}^{(r)}+1}$. Since $\frac{\text{specular}}{\text{scatter}} = \frac{\text{total}-\text{scatter}}{\text{scatter}} = \frac{\text{total}}{\text{scatter}} - 1$, we have:

$$\kappa_{mn}^{(ss)} = \frac{p_{mn}^{(ss)}}{\frac{p_{mn}^{(s)}(d_{mn})}{\kappa_{mn}+1} + \frac{p_{mn}^{(s)}(d_{mn}^{(r)})}{\kappa_{mn}^{(r)}+1}} - 1. \tag{4.20}$$

Next, we derive the average received power $p_{mn}^{(as)}$ for signal propagation from a satellite node in air to a sensor node buried in soil.

4.2.2.2 Multi-media extension: air to soil communication

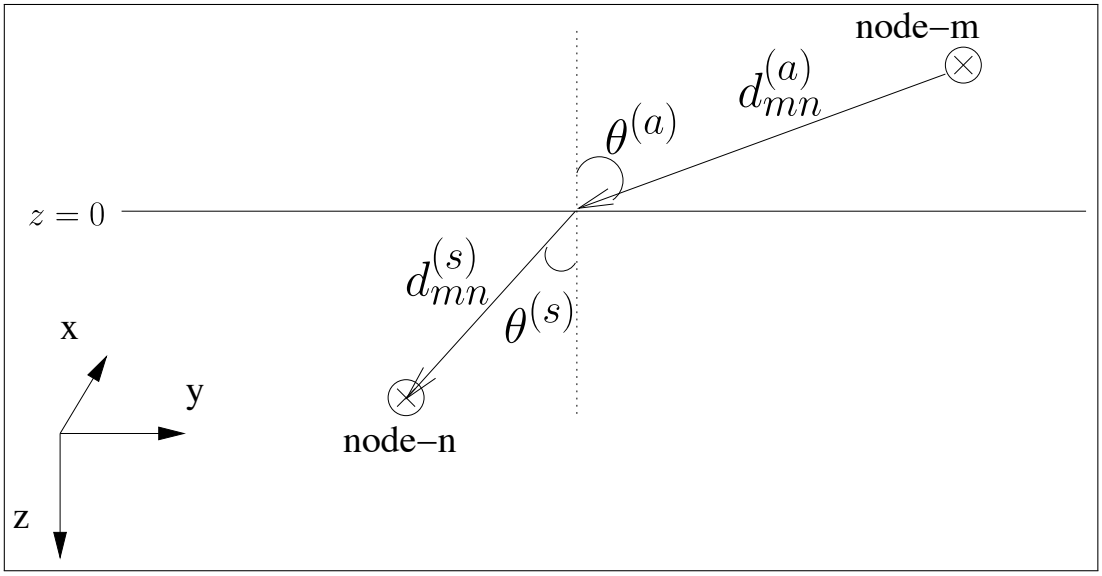


Figure 4.2 Air to soil communication between two nodes m and n.

When the sender node m is in air while the receiver node n is in soil, the average received power is given by:

$$p_{mn}^{(as)} = \eta (d_{mn}^{(a)})^{-k^{(a)}} (d_{mn}^{(s)})^{-k^{(s)}} e^{-2\alpha^{(s)} d_{mn}^{(s)}} \tau, \tag{4.21}$$

where $\tau = 1 - \rho$ is the transmission coefficient, (ρ is given by Equation (4.17)), and $d_{mn}^{(a)}$ and $d_{mn}^{(s)}$ are, respectively, the distances traveled in the air and soil, which are computed as the

solutions of the following two equations:

$$\frac{d_{mn}^{(a)}}{\sqrt{(d_{mn}^{(a)})^2 - z_m^2}} \frac{\sqrt{(d_{mn}^{(s)})^2 - z_n^2}}{d_{mn}^{(s)}} \left(= \frac{\sin\theta^{(a)}}{\sin\theta^{(s)}} \right) = \frac{\lambda^{(a)}}{\lambda^{(s)}} \tag{4.22}$$

$$\sqrt{(d_{mn}^{(a)})^2 - z_m^2} + \sqrt{(d_{mn}^{(s)})^2 - z_n^2} = \sqrt{(x_m - x_n)^2 + (y_m - y_n)^2}, \tag{4.23}$$

where $\lambda^{(a)}$ and $\lambda^{(s)}$ represent the wavelengths in the air and the soil respectively. This result follows from the application of the Snell's law of refraction.

Because of the high refractive index of the soil compared to the air, the signal travels almost vertically downwards in the soil so as to minimize its sojourn time in the soil (this fact is also demonstrated by the small value of the Brewster's angle for the soil-air interface [70]). Accordingly, we can obtain the following approximations:

$$d_{mn}^{(s)} \approx z_n, \tag{4.24}$$

$$d_{mn}^{(a)} \approx \sqrt{(x_m - x_n)^2 + (y_m - y_n)^2 + z_m^2}. \tag{4.25}$$

Thus, we have expressed the parameters of the distribution of the received signal strength for both signal propagation scenarios in our application in terms of the distance and hence, the coordinates of the respective sender and receiver nodes. In the following we formulate the optimization problems for the estimation of the node coordinates given the received signal strength values at each node for signal arriving from all its neighboring nodes and the four satellite nodes.

4.2.3 MLE-based localization

Now, we formulate a maximum likelihood problem to estimate the location coordinates of the nodes given the measurements of the received signal powers at the nodes from their neighboring underground as well as above-ground (satellite) nodes.

Letting N denote the set of node pairs (m, n) that communicate to gather the received signal power measurements, the log-likelihood of all the measured powers for all the sender-receiver

pairs $(m, n) \in N$ can be expressed using Equation (4.2) as follows:

$$L(\boldsymbol{\Theta}|\boldsymbol{R}) = -\sum_{(m,n)\in N} \left\{ \ln(p_{mn}) - \ln(1+\kappa_{mn}) + \left(r_{mn}\frac{\kappa_{mn}+1}{p_{mn}} + \kappa_{mn} \right) - \ln\left[I_0\left(\sqrt{\frac{4(\kappa_{mn}+1)\kappa_{mn}r_{mn}}{\kappa_{mn}}} \right) \right] \right\} \tag{4.26}$$

where $\boldsymbol{R}$ is the set of all measurements of the received signal powers ($\{r_{mn}|(m,n) \in N\}$) and $\boldsymbol{\Theta}$ is the set of unknown parameters to be estimated (i.e., the locations). Note that in Equation (4.26), we have suppressed the superscripts 'as' (air-to-soil) and 'ss' (soil-to-soil) in the parameters p_{mn} and κ_{mn} for the notational simplicity, but those are easily introduced depending on whether the sender node is in the air or the soil (the receiver node is always in the soil). Also note that in Equation (4.26), the received signal power r_{mn} is measured and hence *known*, whereas the set of *unknown* parameters consist of p_{mn} and κ_{mn} which in turn are functions of the unknown location coordinates that must be estimated.

The average powers (p_{mn}'s) are expressed in terms of the location coordinates of the nodes m and n following Equations (4.16 -4.19), while the Rician factors (κ_{mn}'s) are expected to be constants, independent of the locations, over the duration of the round and can be estimated at the beginning of each round [65], [66], [67], [68], [71]. An MLE approach will proceed by writing the parameters p_{mn} and κ_{mn} appearing in the likelihood function of Equation (4.26) as the corresponding functions of the location coordinates of the sensor and satellite nodes as derived in Section 4.2.2, and next optimizing the likelihood function with respect to the location variables.

Alternatively, we can approximate the non-central χ^2 distributions of the likelihood function in the form of simpler Gaussian approximations [72] for performing MLE, or use the approach based on the Method of Moments that avoids solving any optimization problem, and instead reduces the parameter estimation problem to solving a set of algebraic equations in which the first few moments of the distribution at hand are simply equated to their respective sample-based values. We develop these two approaches in the following two subsections.

4.2.3.1 MLE based on Gaussian approximation

From [72], given a set M of independent Gaussian random variables $\{X_m \sim \mathcal{N}(\mu_m, \sigma_m^2)\}_{m\in M}$ so that $R = \sum_{m\in M} \frac{X_m^2}{\sigma_m^2}$ is non-central χ^2 with scale 1, degree M and non-centrality parameter $\gamma = \sum_{m\in M} \frac{\mu_m^2}{\sigma_m^2}$, the random variable $(\sqrt{R} - \sqrt{\gamma}) - \frac{(M-1)(\ln\sqrt{R} - \ln\sqrt{\gamma})}{2(\sqrt{R}-\sqrt{\gamma})}$ is approximately standard normal distributed. Recall that for any sender-receiver pair (m, n), the received signal power R_{mn} with mean p_{mn} and specular-to-scatter ratio κ_{mn} possesses $M = 2$ degrees of freedom due to the in-phase and the quadrature-phase, in which each phase shares half the total scatter power, $\frac{p_{mn}}{2(\kappa_{mn}+1)}$. Consequently the received power R_{mn} normalized with respect to the phase-wise scatter power, namely the power $R'_{mn} := R_{mn} / \left(\frac{p_{mn}}{2(\kappa_{mn}+1)}\right)$, is distributed as a non-central χ^2 with scale 1, degrees of freedom $M = 2$ and non-centrality parameter $\gamma_{mn} = 2\kappa_{mn}$. Accordingly, the random variable $R^{\circ}_{mn} := (\sqrt{R'_{mn}} - \sqrt{\gamma_{mn}}) - \frac{\ln\sqrt{R'_{mn}} - \ln\sqrt{\gamma_{mn}}}{2(\sqrt{R'_{mn}} - \sqrt{\gamma_{mn}})}$ is approximately standard Normal distributed.

Consequently, it follows that the log-likelihood function of the set of random variables $\{R^{\circ}_{mn}, (m, n) \in N\}$ is given by:

$$L(\boldsymbol{\Theta}|\boldsymbol{R}^{\circ}) = -\frac{N}{2}\ln(2\pi) - \sum_{(m,n)\in N} \left\{ \sqrt{\frac{2R_{mn}(\kappa_{mn}+1)}{p_{mn}}} - \sqrt{2\kappa_{mn}} - \frac{\ln\sqrt{\frac{2R_{mn}(\kappa_{mn}+1)}{p_{mn}}} - \ln\sqrt{2\kappa_{mn}}}{2\left(\sqrt{\frac{2R_{mn}(\kappa_{mn}+1)}{p_{mn}}} - \sqrt{2\kappa_{mn}}\right)} \right\}^2, \tag{4.27}$$

where the parameters p_{mn} and κ_{mn} are functions of sender m and receiver n locations, as derived in Section 4.2.2. Given the received signal strength measurements $\boldsymbol{R}$, sensor nodes's coordinates $\boldsymbol{\Theta}$ that maximize this likelihood function are the maximum likelihood estimates.

4.2.3.2 Method of moments

Alternatively, we can use the method of moments technique to estimate the location coordinates of the nodes without having to solve the more complex maximum likelihood optimization problem for the non-central χ^2 received signal strength model. In our setting, following the

properties of non-central χ^2 distribution, the first and second moments of R_{mn} are given by:

$$\mathbb{E}[R_{mn}] = p_{mn}, \tag{4.28}$$

$$\mathbb{E}[R_{mn}^2] = p_{mn}^2 \left(1 + \frac{1 + 2\kappa_{mn}}{(1 + \kappa_{mn})^2}\right). \tag{4.29}$$

These are equated to the corresponding sample-based estimates of the moments, which are then algebraically solved to obtain the estimates of the unknown parameters p_{mn}'s and κ_{mn}'s, and consequently the location coordinates.

4.3 Localization based on time of arrival

The time delay between the transmission of a beacon signal at one node and its reception at another node can be used to estimate the distance between the nodes. This delay is estimated using a matched filter at the receiving node, where a noise-free transmitted signal is available. The copy of the transmitted signal is delayed in time by a certain amount and its correlation with the received signal is measured for a certain observation time. The delay value at which this output is maximum gives an estimate of the propagation delay between the transmitter and receiver. However, this estimate is uncertain owing to noise in the received signal. Furthermore, multi-path interference may cause additional uncertainty in the estimate. For background, we introduce the single-path case first.

4.3.1 Variance of the MLE of ToA for single path

We assume that the transmitted signal $s(t)$ is corrupted by additive white Gaussian noise $n(t)$ upon reception at the receiver node as $r(t)$, after a distance and medium dependent attenuation A and delay τ:

$$r(t) = As(t - \tau) + n(t). \tag{4.30}$$

The auto-covariance function of the noise is $\mathbb{E}[n(t)n(t')] = \frac{N_0}{2}\delta(t - t')$, where $\delta(\cdot)$ is the dirac-delta function and $\frac{N_0}{2}$ is the noise power spectral density.

$r(t)$ is observed over an interval $[0, T]$ at the receiver node. To show that MLE based on ToA reduces to a matched filter described above, we proceed by discretization and then take the limit

to get back to the continuous domain. Accordingly, let us consider the observed data consisting of $n+1$ equally spaced samples of $r(t)$, at time instants $t_k = k\Delta t$,where $t_0 = 0$, $t_n = n\Delta t = T$, and $k \in \{0, 1, 2, ..., n\}$. The individual samples r_k are Gaussian random variables with mean As_k and variance $\frac{N_0}{2}\delta[0] = \frac{N_0}{2}$, where s_k is a sample of the signal $s(t)$ at time $t_k - \tau$, defined as $s_k = s(t_k - \tau)\Delta t$. The observed data-vector $\mathbf{r}$ has a multivariate-Gaussian distribution, where the joint-pdf is driven by the noise pdf. The covariance matrix Φ of $\mathbf{r}$ is $(n+1)$ by $(n+1)$, with $\{j, k\}^{th}$ element:

$$\phi_{jk} = \frac{N_0}{2}\delta[j-k], \tag{4.31}$$

where ϕ_{jk} is the covariance between r_j and r_k. The mean of $\mathbf{r}$ equals the samples of the transmitted signal:

$$E[\mathbf{r}] = A\mathbf{s} := A[s_0 \dots s_k \dots s_n]^T, \tag{4.32}$$

where ^{T} denotes vector transpose operation. Thus, the joint-pdf of $\mathbf{r}$ is:

$$p(\mathbf{r};\tau) = \frac{1}{(2\pi)^{\frac{(n+1)}{2}}|\Phi|^{\frac{1}{2}}} \exp\left\{\frac{-1}{2}(\mathbf{r}-A\mathbf{s})^T\Phi^{-1}(\mathbf{r}-A\mathbf{s})\right\}, \tag{4.33}$$

$$= \frac{1}{(\pi N_0)^{\frac{(n+1)}{2}}} \exp\left\{-\sum_{k=1}^{n+1}\frac{(r_k - As_k)^2}{N_0}\right\}, \tag{4.34}$$

where Equation (4.34) follows from (4.33) by using (4.31), and substituting $\phi_{jk} = \frac{N_0}{2}\delta[j-k]$ for additive white Gaussian noise.

Noting that the squared terms r_k^2 and $(As_k)^2$ sum up to received and attenuated signal powers, which are independent of the delay of communication, the log-likelihood function, ignoring the terms that do not depend on τ, is given by:

$$\log\Lambda(\tau|\mathbf{r}) = \sum_{k=1}^{n+1}\frac{2r_kAs_k}{N_0}.. \tag{4.35}$$

Under the limit as $\Delta t \to 0$ (and $n \to \infty$), Equation (4.35) becomes:

$$\log\Lambda(\tau|r(t)) = \int_0^T \frac{2}{N_0}r(t)As(t-\tau)dt. \tag{4.36}$$

Thus, the log-likelihood function reduces to the correlator function between $r(t)$ and $s(t)$ given a certain delay between the two, which is the output of a matched filter. Hence, $\hat{\tau}$ that

maximizes this correlator function between the transmitted and received signals turns out to be the maximum likelihood estimate of the delay τ.

The mean of $\hat{\tau}$, denoted $\bar{\tau}$, is determined by the distance between the sender and receiver nodes and the propagation speed of the signal in the medium it travels. The variance of $\hat{\tau}$, being an MLE, equals the Cramer-Rao lower bound (CRLB) and is the inverse of the Fisher information matrix, given by [73]:

$$\sigma^2 = \left[\mathbb{E}\left\{\frac{\partial}{\partial\tau}\left[\log\Lambda(\tau|r(t))\right]\frac{\partial}{\partial\tau}\left[\log\Lambda(\tau|r(t))\right]\right\}\right]^{-1} \tag{4.37}$$

The next expression follows from the derivation given in [73, p. 264-265].

$$\sigma^2 = \left\{\frac{2A^2}{N_0}\int_0^T \frac{\partial}{\partial\tau}s(t-\tau)\frac{\partial}{\partial\tau}s(t-\tau)dt\right\}^{-1} \tag{4.38}$$

$$= \left\{\frac{2A^2}{N_0}\int_0^T [s'(t)]^2 dt\right\}^{-1} \tag{4.39}$$

$$= \left\{\frac{2A^2}{N_0}\int_0^T (2\pi f)^2|S(f)|^2 df\right\}^{-1} \tag{4.40}$$

$$= \left\{8\pi^2\frac{p}{N_0}T\beta^2\right\}^{-1}, \tag{4.41}$$

$$= \left\{8\pi^2 TB\beta^2\text{SNR}\right\}^{-1} \tag{4.42}$$

where $S(f)$ is the Fourier transform of the signal $s(t)$, and A^2 is the signal power attenuation due to propagation in the respective medium, so that $p = \frac{A^2\int|S(f)|^2df}{T}$ is received signal-power, $\beta = \sqrt{\frac{\int_0^T f^2|S(f)|^2df}{\int_0^T|S(f)|^2df}}$ is a function of the signal, B is the bandwidth of the signal and $\text{SNR} = \frac{p}{N_0B}$ is the signal to noise power ratio. Thus, we have established that the variance of the time of arrival as estimated by the receiver's matched filter, which is also the maximum likelihood estimate, is given by Equation (4.41). Since the MLE is asymptotically normal, the estimated time of arrival $\hat{\tau}$ can be approximated to be a Gaussian distributed random variable:

$$p_{\hat{\tau}}(\tau) = \frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{(\tau-\bar{\tau})^2}{2\sigma^2}}. \tag{4.43}$$

In the next section we derive the relations that establish the dependence of $\bar{\tau}$ and σ^2 on the location coordinates of the sensor nodes for our precision agriculture application. In this way, the pdf of $\hat{\tau}$ can be parametrized in terms of the location coordinates in order to estimate the latter from the observed times of arrival for signals transmitted between neighboring pairs

of sensor nodes and between the sensor nodes and the satellite nodes with known location coordinates.

4.3.2 Mean and variance of time of arrival in terms of location coordinates

4.3.2.1 Multi-path extension: soil to soil communication

As discussed in Section 4.2.2.1 and Figure 4.1, a signal transmitted by an underground sensor node arrives at another underground sensor node via two paths; one along the direct line-of-sight path and another along a path after reflection from the ground surface. The received signal in time domain is given by:

$$r(t) = As(t-\tau) + A^r s(t-\tau^r) + n(t), \tag{4.44}$$

where A and A^r are the attenuations along the two paths and τ and τ^r are the propagation delays along the two paths.

Following the steps in Section 4.3.1 for the received signal given by Equation (4.44), the joint log-likelihood function for the parameter $\boldsymbol{\tau} = [\tau \quad \tau^r]^T$ is given by:

$$\log \Lambda(\boldsymbol{\tau}|r(t)) = \int_0^T \frac{2}{N_0} r(t) \left\{ As(t-\tau) + A^r s(t-\tau^r) \right\} dt. \tag{4.45}$$

The estimate of $\boldsymbol{\tau}$ that maximizes Equation (4.45) is the maximum likelihood estimate. Hence, the covariance matrix of the estimate is given by the inverse of the Fisher information matrix. The $(i,j)^{\text{th}}$ element of the covariance matrix $\Sigma_{2\times 2}$ is, then, given by:

$$\sigma_{ij} = \left[\mathbb{E} \left\{ \frac{\partial}{\partial \theta_i} \left[\log \Lambda(\boldsymbol{\tau}|r(t)) \right] \frac{\partial}{\partial \theta_j} \left[\log \Lambda(\boldsymbol{\tau}|r(t)) \right] \right\} \right]^{-1}, \tag{4.46}$$

where $i, j \in \{1, 2\}$ correspond to the line-of-sight and reflected paths, $\theta_1 = \tau$ and $\theta_2 = \tau^r$.

The next expression follows from the derivation given in [73, p. 264-265].

$$\sigma_{ij} = \left\{ \frac{2A_iA_j}{N_0} \int_0^T \frac{\partial}{\partial\theta_i} s(t-\theta_i) \frac{\partial}{\partial\theta_j} s(t-\theta_j) dt \right\}^{-1} \tag{4.47}$$

$$= \left\{ \frac{2A_iA_j}{N_0} \int_0^T [s'(t)]^2 dt \right\}^{-1} \tag{4.48}$$

$$= \left\{ \frac{2A_iA_j}{N_0} \int_0^T (2\pi f)^2 |S(f)|^2 df \right\}^{-1} \tag{4.49}$$

$$= \left\{ 8\pi^2 \frac{\sqrt{p_i}\sqrt{p_j}}{N_0} T\beta^2 \right\}^{-1}, \tag{4.50}$$

$$= \left\{ 8\pi^2 TB\beta^2 \sqrt{\mathrm{SNR}_i}\sqrt{\mathrm{SNR}_j} \right\}^{-1}, \tag{4.51}$$

where $\mathrm{SNR}_i = \frac{p_i}{N_0 B}$ and $\mathrm{SNR}_j = \frac{p_j}{N_0 B}$ are the signal to noise power ratios for the two paths i and j. Note that for a sender-receiver pair (m, n), $p_1 = p_{mn}$ and $p_2 = p^r_{mn}$, are obtained using Equation (4.15), with the latter containing an additional multiplicative factor of reflection constant ρ modeling the power loss due to reflection from the ground surface. ρ is given by Equation (4.17). Thus, we have $p_1 = p_{mn} = \eta(d_{mn})^{-k_s} e^{-2\alpha_s d_{mn}}$ and $p_2 = p^r_{mn} = \eta(d_{mn})^{-k_s} e^{-2\alpha_s d_{mn}} \rho$.

For sender-receiver pair (m, n), the mean value $\bar{\boldsymbol{\tau}}_{mn} = [\bar{\tau}_{mn} \quad \bar{\tau}^r_{mn}]^T$ is determined by the propagation distances and the speed of the signal:

$$\bar{\tau}_{mn} = \frac{d_{mn}}{c_s}, \tag{4.52}$$

$$\bar{\tau}^r_{mn} = \frac{d^{(r)}_{mn}}{c_s}, \tag{4.53}$$

where the line of sight propagation distance d_{mn} is given by Equation (4.18), the reflected path propagation distance $d^{(r)}_{mn}$ is given by Equation (4.19) and the speed of light in soil c_s is given by:

$$c_s = \frac{1}{\sqrt{\frac{\mu_s \epsilon'_s}{2}\left(\sqrt{1 + \left(\frac{\sigma_s}{\omega\epsilon'_s}\right)^2} + 1\right)}}, \tag{4.54}$$

where ϵ'_s, μ_s and σ_s are, respectively, the real part of permittivity, permeability and the conductivity of soil.

Following the same argument as in Section 4.3.1, $\hat{\boldsymbol{\tau}}$ is asymptotically distributed as a bivariate Gaussian random variable with the pdf:

$$p_{\hat{\boldsymbol{\tau}}_{mn}}(\boldsymbol{\tau}_{mn}) = \frac{1}{(2\pi)|\Sigma_{mn}|^{1/2}} \exp\left\{ -\frac{1}{2} (\boldsymbol{\tau}_{mn} - \bar{\boldsymbol{\tau}}_{mn})^T \Sigma^{-1}_{mn} (\boldsymbol{\tau}_{mn} - \bar{\boldsymbol{\tau}}_{mn}) \right\}. \tag{4.55}$$

4.3.2.2 Multi-media extension: air to soil communication

For signal propagation between a satellite node m and a sensor node n, the propagation medium is partly air and partly soil, see Figure 4.2, where $d_{mn}^{(a)}$ and $d_{mn}^{(s)}$, the propagation distances in air and soil, respectively, were derived earlier in Equations (4.22), (4.23), (4.24) and (4.25).

Therefore, we obtain an expression for the mean value of the propagation delay between the satellite and sensor nodes:

$$\overline{\tau}_{mn} = \frac{d_{mn}^{(a)}}{c_a} + \frac{d_{mn}^{(s)}}{c_s}, \tag{4.56}$$

where $c_a \approx c$ is the signal speed in air that approximately equals the speed of light in vacuum, and c_s is given by Equation (4.54).

Thus, from Equation (4.56) we obtain an expression for the mean of the time of arrival in terms of the location coordinates of the sender and receiver nodes, as the first step in expressing its pdf parameterized by the location coordinates.

Next, we obtain an expression for the variance in terms of the location coordinates. From Equation (4.41):

$$\sigma_{mn}^2 = \left\{ 8\pi^2 \frac{p_{mn}^{(as)}}{N_0} T\beta^2 \right\}^{-1} = \left\{ 8\pi^2 TB\beta^2 \mathrm{SNR}_{mn} \right\}^{-1}, \tag{4.57}$$

where $p_{mn}^{(as)}$, the average received power for air to soil communication, was derived earlier in Equation (4.21) in terms of the location coordinates of the satellite node m and sensor node n.

4.3.3 MLE-based localization

Next we formulate a maximum likelihood problem to estimate the location coordinates of the nodes given the measurements of the times of arrivals at the nodes from their neighboring underground as well as above-ground (satellite) nodes.

Let N_{ss} and N_{as} denote the set of soil-to-soil and air-to-soil node pairs that communicate to gather the time of arrival data for soil to soil signal propagation and air to soil propagation, respectively. The log-likelihood of the time of arrival distribution parameters for all the sender-

receiver pairs $(m,n) \in N_{ss} \cup N_{as}$ can be expressed as follows:

$$L(\boldsymbol{\Theta}|\boldsymbol{T}) = -\sum_{(m,n)\in N_{as}} \left\{ \ln(\sigma_{mn}) + \frac{(\tau_{mn} - \overline{\tau}_{mn})^2}{2\sigma^2_{\hat{\tau}_{mn}}} \right\} - \sum_{(m,n)\in N_{ss}} \left\{ \frac{1}{2}\ln|\Sigma_{mn}| + \frac{1}{2}(\boldsymbol{\tau}_{mn} - \bar{\boldsymbol{\tau}}_{mn})^T \Sigma^{-1}_{mn}(\boldsymbol{\tau}_{mn} - \bar{\boldsymbol{\tau}}_{mn}) \right\}, \tag{4.58}$$

where $\boldsymbol{T}$ represents the time of arrival data $\{\tau_{mn}|(m,n) \in N_{as}, \boldsymbol{\tau}_{mn} = [\tau_{mn} \quad \tau^r_{mn}]^T|(m,n) \in N_{ss}\}$ between all pairs of nodes that communicate with each other to gather localization data. σ_{mn}, $\overline{\tau}_{mn}$, Σ_{mn} and $\bar{\boldsymbol{\tau}}_{mn} = [\bar{\tau}_{mn} \quad \bar{\tau}^r_{mn}]^T$ were expressed in terms of the location coordinates of the sensor nodes and the known location coordinates of the satellite nodes in Sections 4.3.2.1 and 4.3.2.2 for the different signal propagation scenarios in our application. Thus, the maximum likelihood estimates of the location coordinates are obtained as the values that maximize the log likelihood function given by Equation (4.58).

4.3.3.1 Method of moments

Alternatively, we can use the method of moments technique to estimate the location coordinates of the nodes without having to solve the more complex maximum likelihood optimization problem. In our setting, following the properties of Gaussian distribution, the first moment of τ_{mn} equals $\overline{\tau}_{mn}$, whereas its second moment is given by σ^2_{mn} for air to soil signal propagation. For soil to soil propagation, the first two moments are given by $\bar{\boldsymbol{\tau}}_{mn}$ and Σ_{mn}, respectively. These can be equated to the corresponding sample-based estimates of the moments, which can then be algebraically solved to obtain the estimates of the unknown parameters σ_{mn}, $\overline{\tau}_{mn}$, Σ_{mn} and $\bar{\boldsymbol{\tau}}_{mn}$ and consequently, the location coordinates of the sensor nodes.

4.4 Simulation results

We simulate a network of 25 sensor nodes in a square field of size 150 m $\times$150 m to validate and evaluate our localization framework. The sensor nodes are randomly deployed within the square field at a depth between $3-5$ cm. Both localization models discussed in Sections 4.2 and 4.3 depend on the soil permittivities which in turn are functions of the moisture content of the soil. We assume a clay loam soil with a clay content of 20%. For this soil type, the relative real

Fractional volume of water	ϵ_s'	ϵ_s''
0%	2.36	0.0966
10%	5.086	0.441
30%	16.410	2.368
40%	24.485	3.857

Table 4.1 Real and imaginary parts of relative permittivity of soil for different moisture contents.

Parameter	Symbol	Value
Transmit power (Satellite node)	$P^{(a)}$	30 dBm
Transmit power (Sensor node)	$P^{(s)}$	30 dBm
Thermal noise	N_0	-110 dBm
Path loss factor (air)	k_a	2
Path loss factor (soil)	k_s	2
Relative permeability (soil)	μ_s	1.0084
Frequency	f	433 MHz
Wavelength	λ	0.7 m
Number of readings per node pair (RSS)	N_{RSS}	100
Number of readings per node pair (ToA)	N_{ToA}	100
Signal duration	T_S	1ms

Table 4.2 Parameters used in localization simulations.

and imaginary values of the permittivity are computed [74], [75], [76], [77] for various values of volumetric water contents as given in Table 4.1. Other parameters used in the simulations are summarized in Table 4.2. The relative magnetic permeability of soil is close to one for soils not containing significant iron [78]. Thermal noise is assumed to be -110 dBm based on standard receiver sensitivity, which is also the case for our receivers [79]. Thus, applying Equations (4.11), (4.15), (4.16) and (4.21), with the parameters of Tables 4.1 and 4.2, the underground transmission range is approximately 34 m for dry soil at a transmission power level of 30 dBm leading to an average of 3 neighbors per sensor node, whereas the air–to-soil transmission range is approximately 1000 m.

The simulations for data generation and optimization were implemented in Python [30]. SciPy's [80] optimization package was used for implementing the localization estimation. All figures in this chapter have been generated using Matplotlib [81].

We use a hierarchical and iterative approach for estimating the location coordinates of the sensor nodes, applying first the air-to-soil ranging models on the signals transmitted between

the satellite nodes and sensor nodes to arrive at a first set of estimates, followed by the combination of air-to-soil and soil-to-soil models, adding signal data among neighboring sensor nodes to arrive at more refined and accurate estimates of the coordinates. The second stage is applied iteratively with each new iteration initialized with estimates obtained from the preceding iteration, until the estimates converge within a tolerance. The convergence tolerance for X and Y estimates is chosen to be 5 cm, while for Z estimate it is chosen to be 1 cm. Figures 4.3, 4.4, 4.5, 4.6 and 4.7 present the localization estimates obtained using the received signal strength model for five different values of the Rician K-factor (κ) [82], [83]. Figure 4.8 presents the localization estimates obtained using the time of arrival localization model. Each simulation for estimating the nodes' coordinates for the network of chosen size took about 30 minutes on a laptop computer with a 2.4 GHz dual-core processor and 8 GB 1067 MHz DDR3 DRAM. A visual inspection of the results shows that our method successfully estimates the location coordinates of the sensor nodes, with the error margins as described below.

We use a minimal setup of our network to perform variance analysis of our location estimates: three sensor nodes deployed randomly within the field but within the radio communication ranges of each other and four satellite nodes so that we can closely capture the real world deployment scenario where each sensor node has $2-3$ neighboring sensor nodes. Then, we compute the sample standard deviation of the location estimate of one of the sensor nodes. Figure 4.9 shows the sample standard deviation of the estimates for a sample size of 1000 for received signal strength based localization, which we choose as a metric for the error margin of the estimates. X and Y coordinates are estimated with a standard deviation of 0.4 m at a Rician K-factor (κ) level of 20. However, as κ increases to 100, the standard deviation reduces to 0.2 m. Similarly, the standard deviation of the Z coordinate estimate decreases with an increase in κ. The standard deviation in X and Y estimates is relatively insensitive to change in the soil permittivity. However, the Z estimate has the least standard deviation for $\epsilon_s' = 24.5$ and $\epsilon_s'' = 3.86$, corresponding to a moisture content of 40%. The standard deviation of the Z coordinate estimate decreases with the increase in the moisture content of the soil. This behavior can be explained as follows: with the increase in the moisture content the attenuation of the signal increases, decreasing the average received power. From Equations (4.28) and (4.29),

the variance of the received signal strength also reduces, reducing the variance of the estimate.

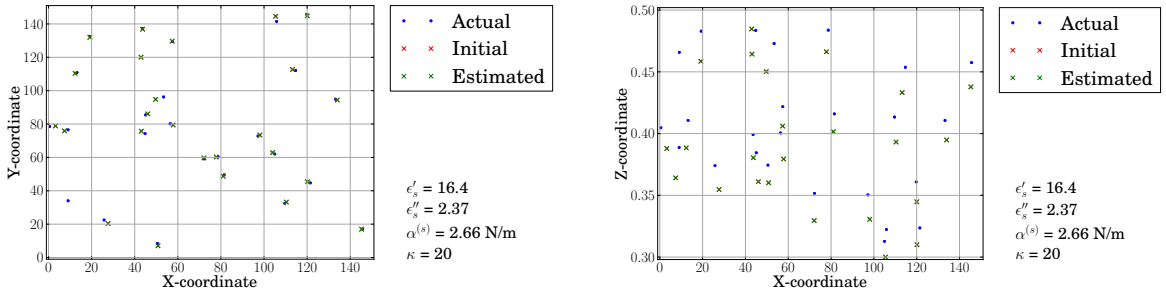
Figure 4.10 shows the sample standard deviation of the estimates for a sample size of 1000 for time of arrival based localization. Time of arrival based localization has almost ten orders of magnitude better error performance than received signal strength based localization. The increased accuracy of time of arrival based localization comes at the cost of clock synchronization among all the nodes in the network. But clock synchronization is built into our network as it is also needed for scheduling (see Section 2.4.2). X and Y coordinate estimates have a standard deviation of about 10^{-9} m while the corresponding quantity for the Z coordinate estimate is 10^{-10} m. The effect of the moisture content is also apparent in time of arrival based localization with the estimates having the least standard deviation for a moisture content of 40%.

The simulation results demonstrate that our technique is efficient, scalable and reliable. Although the results are presented for a field size of 150 m $\times$ 150 m, the technique is equally applicable to larger fields.

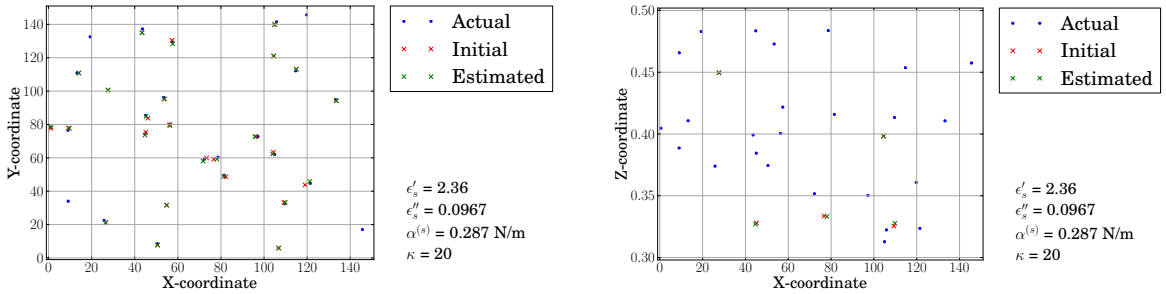
4.4.1 Sensitivity analysis

Sensitivity analysis is used to determine the level of errors that may be introduced due to uncertainty in various parameters employed in the localization models. As an example, the soil permittivity is a parameter of our ranging models used in localization. Hence, localization performance depends on the accuracy with which the soil permittivity values (ϵ_s' and ϵ_s'') are measured by our soil sensors. The uncertainty in their measurement may cause the mean of the estimators to be shifted relative to the actual coordinate values. We study the change in the estimated coordinates as a function of the difference between the measured μ_s, ϵ_s' and ϵ_s'' and their true values from Tables 4.1 and 4.2. We use the same setup for sensitivity analysis as used in variance analysis. That is, a set of three sensor nodes are randomly deployed in the sensor field within the radio communication ranges of each other along with the four satellite nodes.

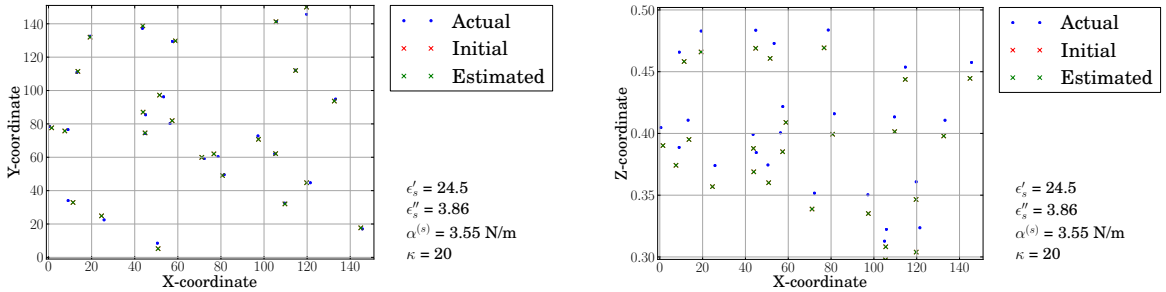
Figures 4.11, 4.12 and 4.13 present the sensitivities of the estimates (the slopes of the estimate with respect to parameters of interest), based on received signal strength localization



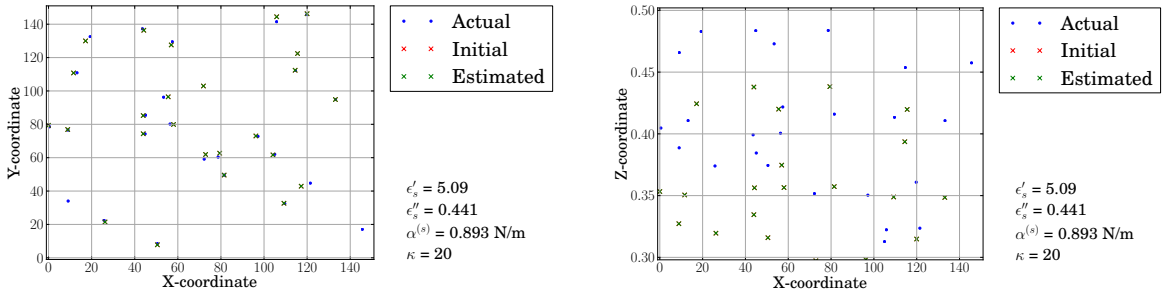
(a) Localization for parameters ($\epsilon'_s = 16.4, \epsilon''_s = 2.37$).



(b) Localization for parameters ($\epsilon'_s = 2.36, \epsilon''_s = 0.0967$)

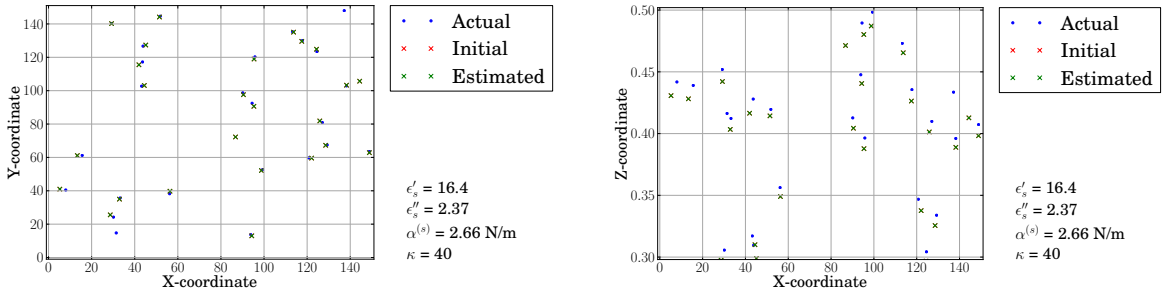


(c) Localization for parameters ($\epsilon'_s = 24.5, \epsilon''_s = 3.86$)

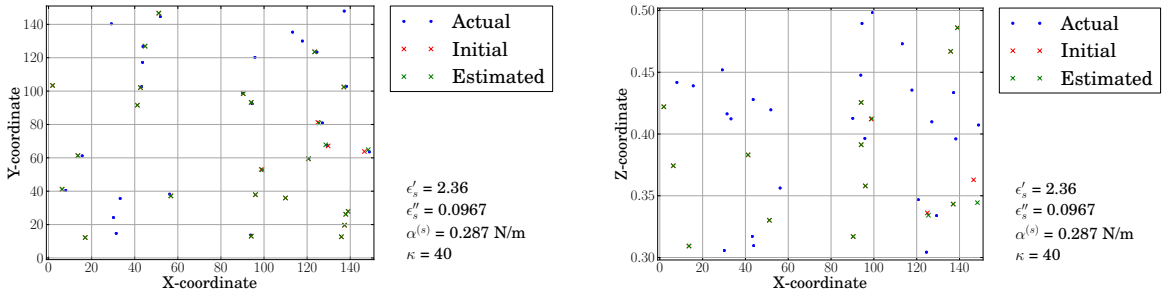


(d) Localization for parameters ($\epsilon'_s = 5.09, \epsilon''_s = 0.441$)

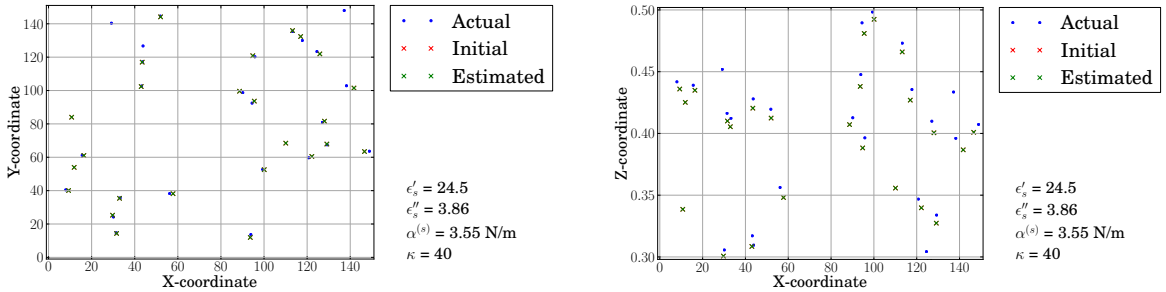
Figure 4.3 Localization using received signal strength model for Rician factor $\kappa = 20$.



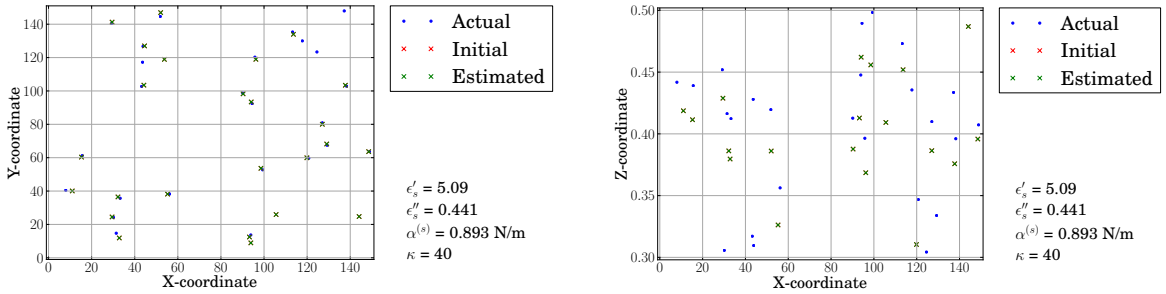
(a) Localization for parameters ($\epsilon'_s = 16.4, \epsilon''_s = 2.37$).



(b) Localization for parameters ($\epsilon'_s = 2.36, \epsilon''_s = 0.0967$)

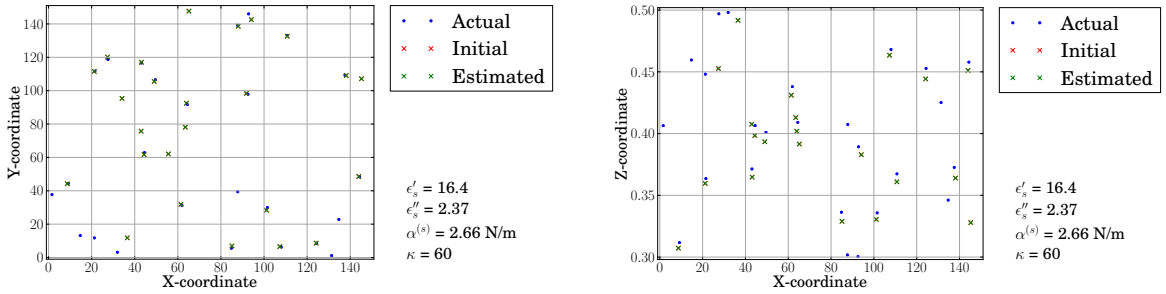


(c) Localization for parameters ($\epsilon'_s = 24.5, \epsilon''_s = 3.86$)

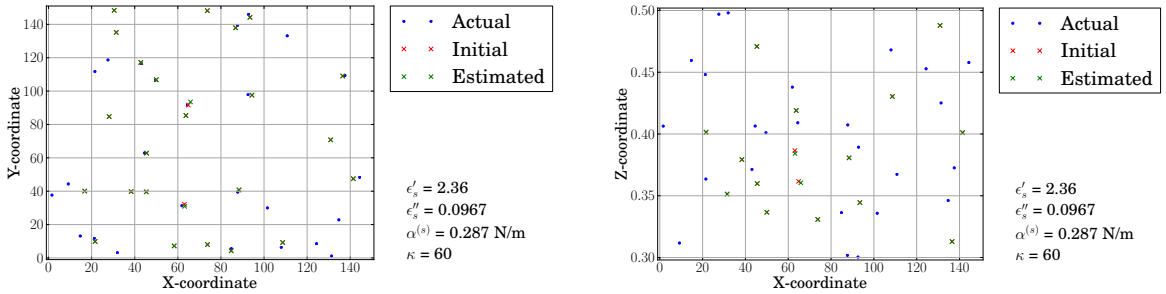


(d) Localization for parameters ($\epsilon'_s = 5.09, \epsilon''_s = 0.441$)

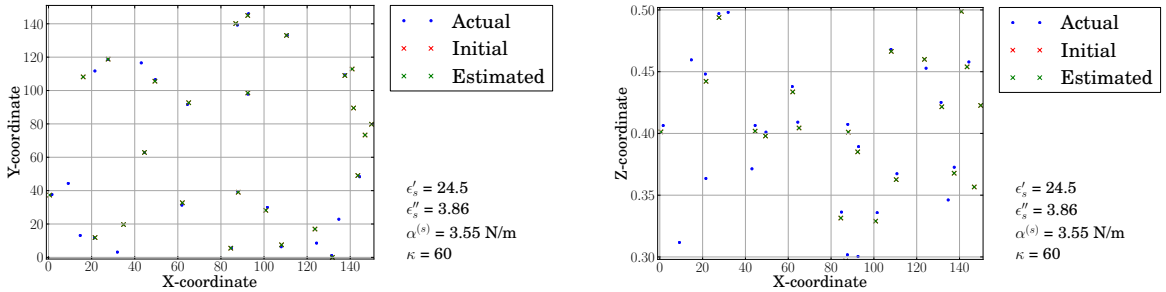
Figure 4.4 Localization using received signal strength model for Rician factor $\kappa = 40$.



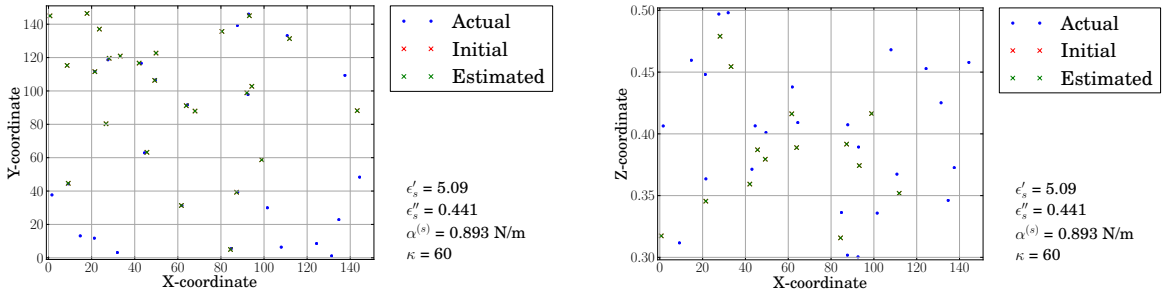
(a) Localization for parameters ($\epsilon'_s = 16.4, \epsilon''_s = 2.37$).



(b) Localization for parameters ($\epsilon'_s = 2.36, \epsilon''_s = 0.0967$)

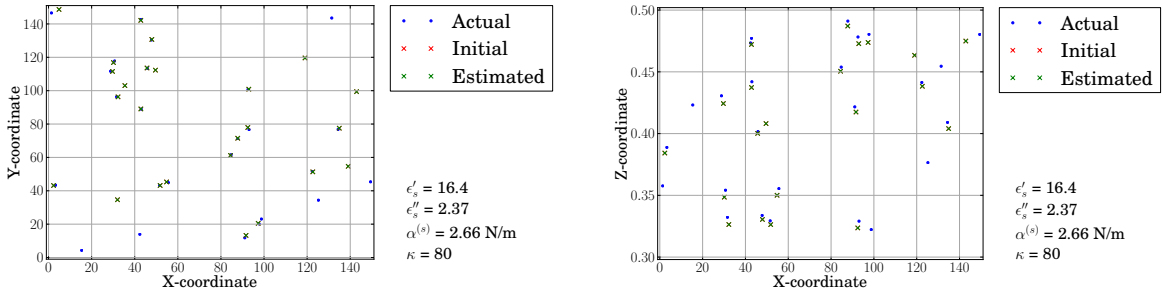


(c) Localization for parameters ($\epsilon'_s = 24.5, \epsilon''_s = 3.86$)

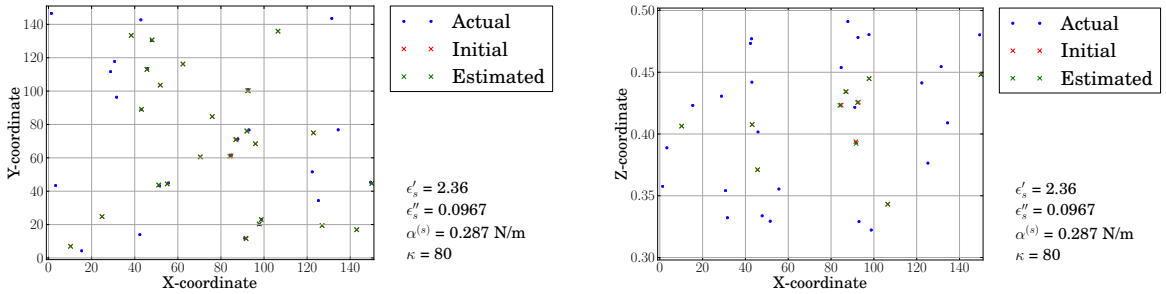


(d) Localization for parameters ($\epsilon'_s = 5.09, \epsilon''_s = 0.441$)

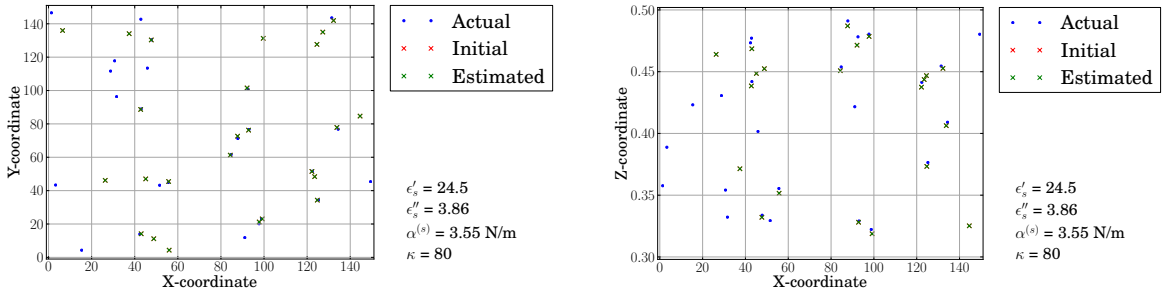
Figure 4.5 Localization using received signal strength model for Rician factor $\kappa = 60$.



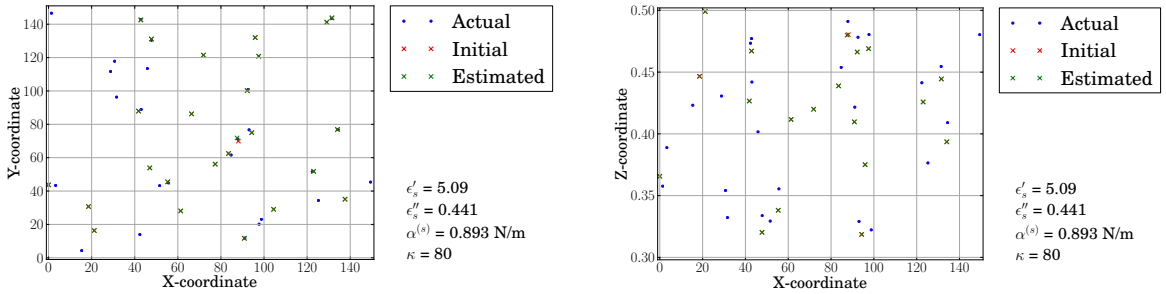
(a) Localization for parameters ($\epsilon'_s = 16.4, \epsilon''_s = 2.37$).



(b) Localization for parameters ($\epsilon'_s = 2.36, \epsilon''_s = 0.0967$)

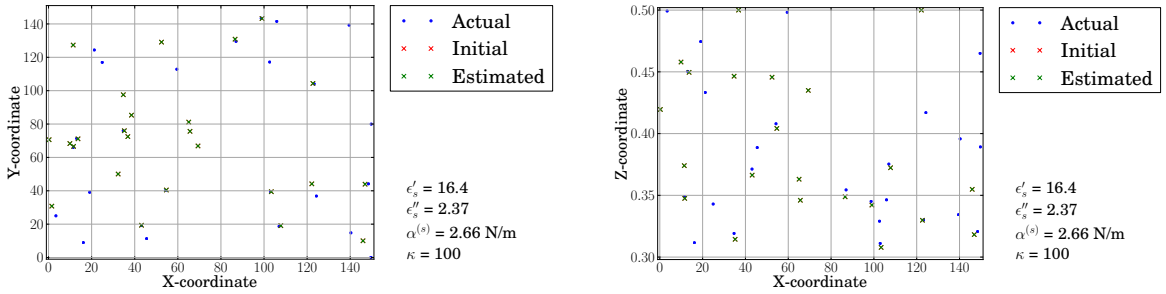


(c) Localization for parameters ($\epsilon'_s = 24.5, \epsilon''_s = 3.86$)

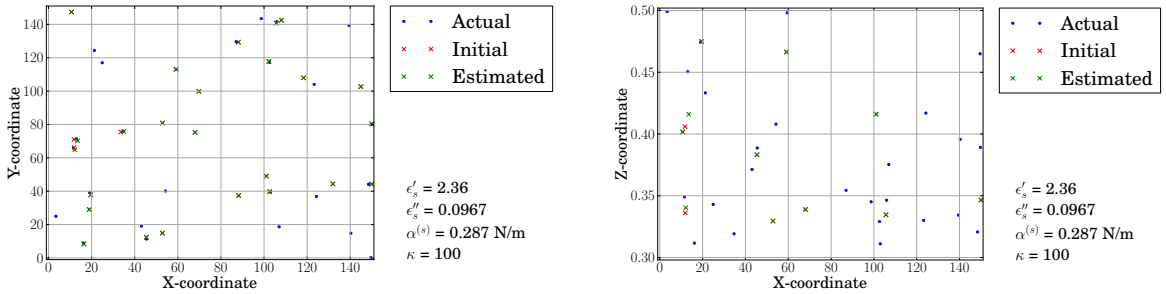


(d) Localization for parameters ($\epsilon'_s = 5.09, \epsilon''_s = 0.441$)

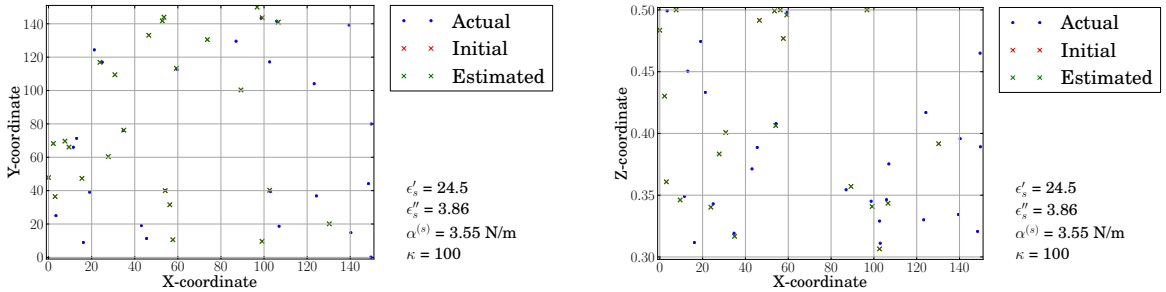
Figure 4.6 Localization using received signal strength model for Rician factor $\kappa = 80$.



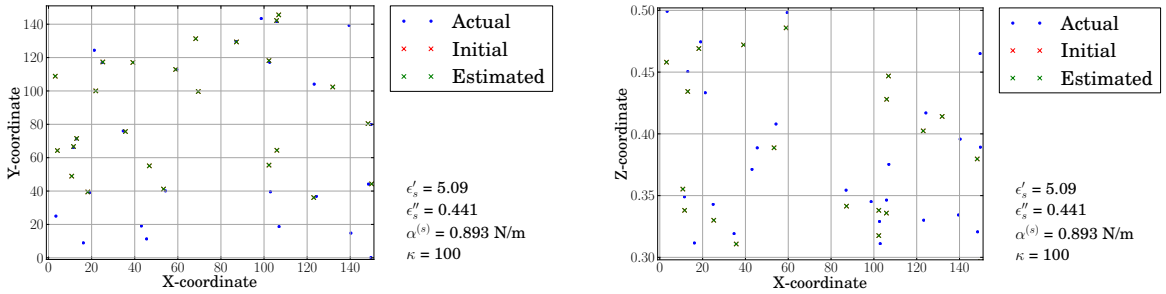
(a) Localization for parameters ($\epsilon'_s = 16.4, \epsilon''_s = 2.37$).



(b) Localization for parameters ($\epsilon'_s = 2.36, \epsilon''_s = 0.0967$)

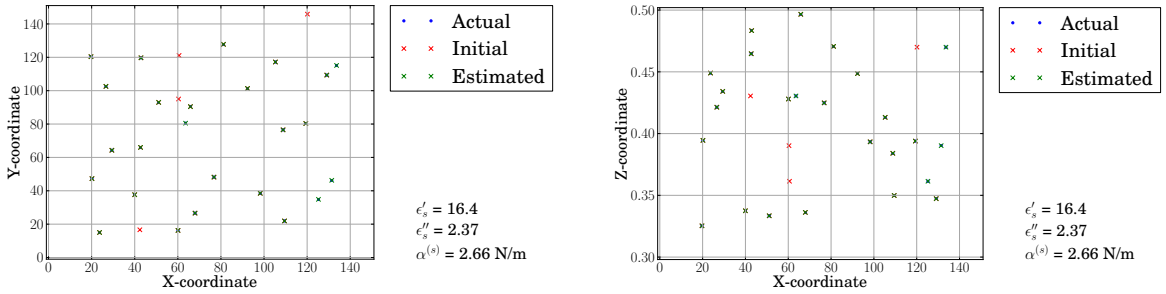


(c) Localization for parameters ($\epsilon'_s = 24.5, \epsilon''_s = 3.86$)

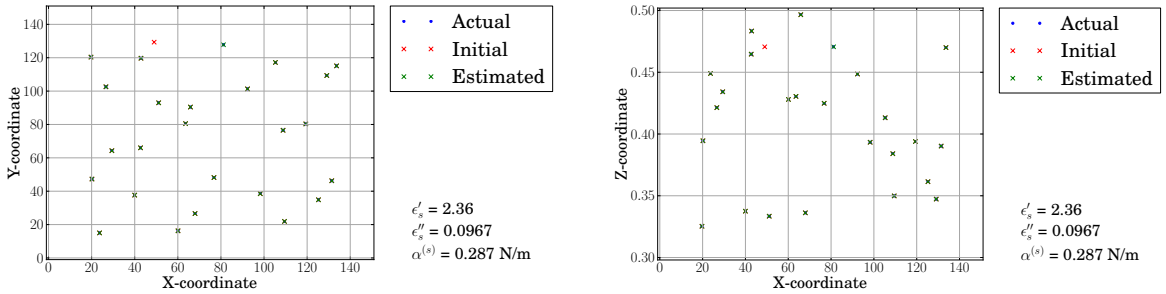


(d) Localization for parameters ($\epsilon'_s = 5.09, \epsilon''_s = 0.441$)

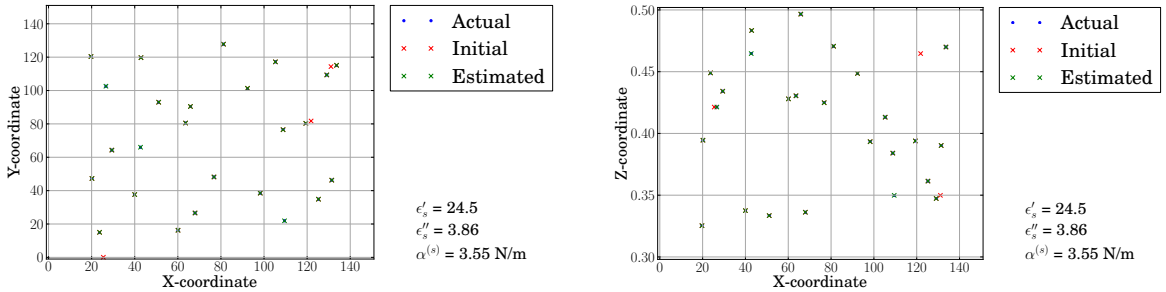
Figure 4.7 Localization using received signal strength model for Rician factor $\kappa = 100$.



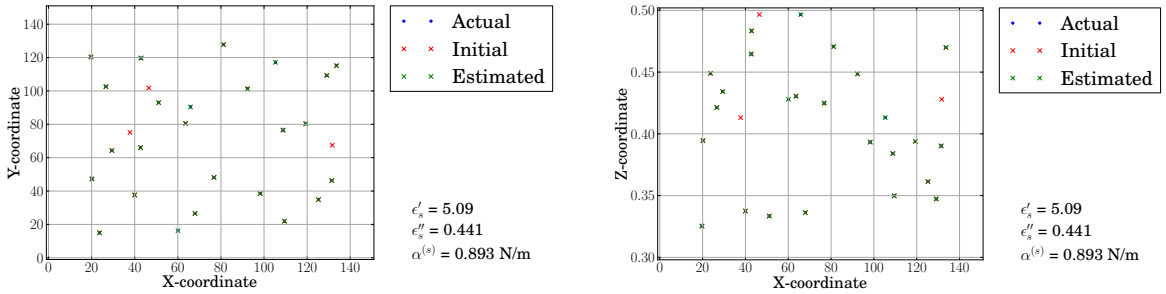
(a) Localization for parameters ($\epsilon'_s = 16.4, \epsilon''_s = 2.37$).



(b) Localization for parameters ($\epsilon'_s = 2.36, \epsilon''_s = 0.0967$)

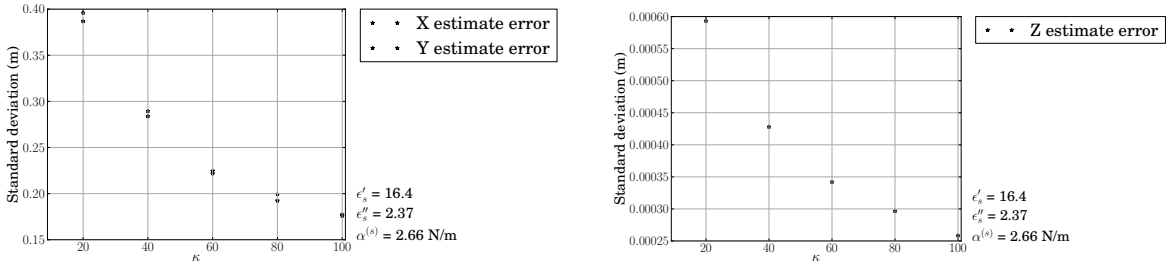


(c) Localization for parameters ($\epsilon'_s = 24.5, \epsilon''_s = 3.86$)

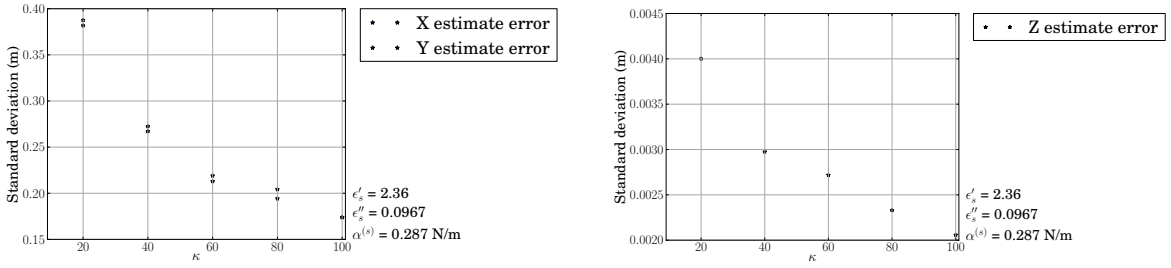


(d) Localization for parameters ($\epsilon'_s = 5.09, \epsilon''_s = 0.441$)

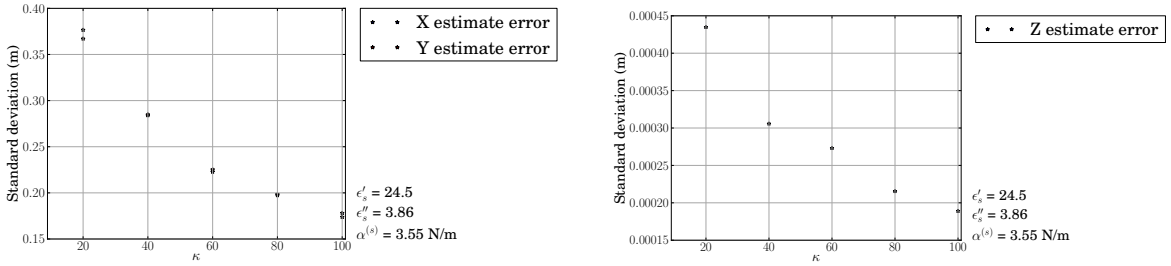
Figure 4.8 Localization using time of arrival model.



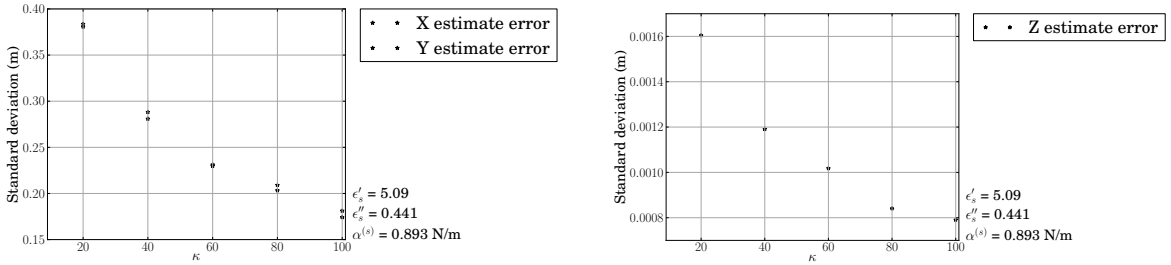
(a) Standard deviation of estimates ($\epsilon'_s = 16.4, \epsilon''_s = 2.37$).



(b) Standard deviation of estimates ($\epsilon'_s = 2.36, \epsilon''_s = 0.0967$)



(c) Standard deviation of estimates ($\epsilon'_s = 24.5, \epsilon''_s = 3.86$)



(d) Standard deviation of estimates ($\epsilon'_s = 5.09, \epsilon''_s = 0.441$)

Figure 4.9 Sample standard deviation of estimates for RSS localization. Sample size = 1000.

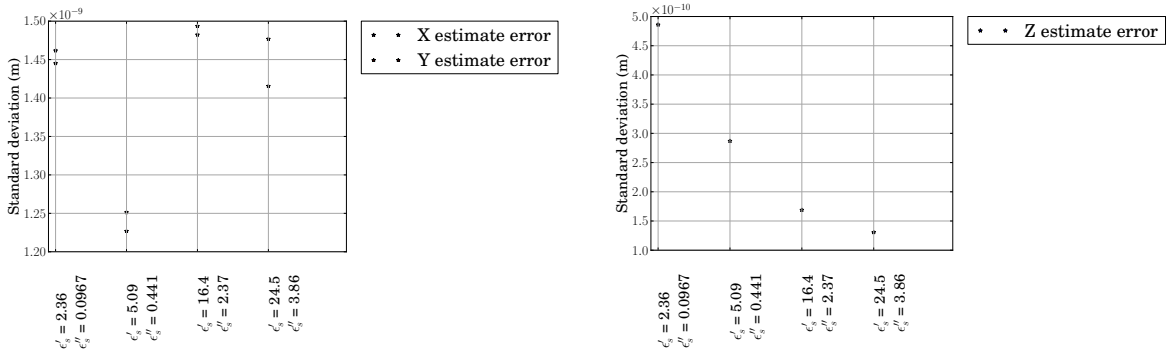


Figure 4.10 Sample standard deviation of estimates for ToA localization. Sample size = 1000.

to ϵ_s', ϵ_s'' and μ_s, respectively; while Figures 4.14, 4.15 and 4.16 present the sensitivities of the estimates based on time of arrival localization to ϵ_s', ϵ_s'' and μ_s, respectively.

Received signal strength localization based estimates are least sensitive to all the parameters studied for dry soil. However, the sensitivities with respect to the different parameters vary. A 10% error in the measured ϵ_s'' caused the largest shift in the estimated location of 0.024 m; whereas the same error in ϵ_s' or μ_s causes a shift of 0.012 m.

Time of arrival localization based estimate sensitivities with respect to ϵ_s' and μ_s are independent of the soil moisture content. However, the sensitivity with respect to ϵ_s'' varies with the moisture content, the estimate being least sensitive to change in ϵ_s'' for dry soil. The estimate is less sensitive to change in ϵ_s'' when compared with the sensitivity with respect to ϵ_s' and μ_s, with a 10% error in the parameter causing a shift of only 0.000035 m in the estimate. Equivalent errors in ϵ_s' and μ_s cause shifts of 0.0025 m and 0.0007 m, respectively.

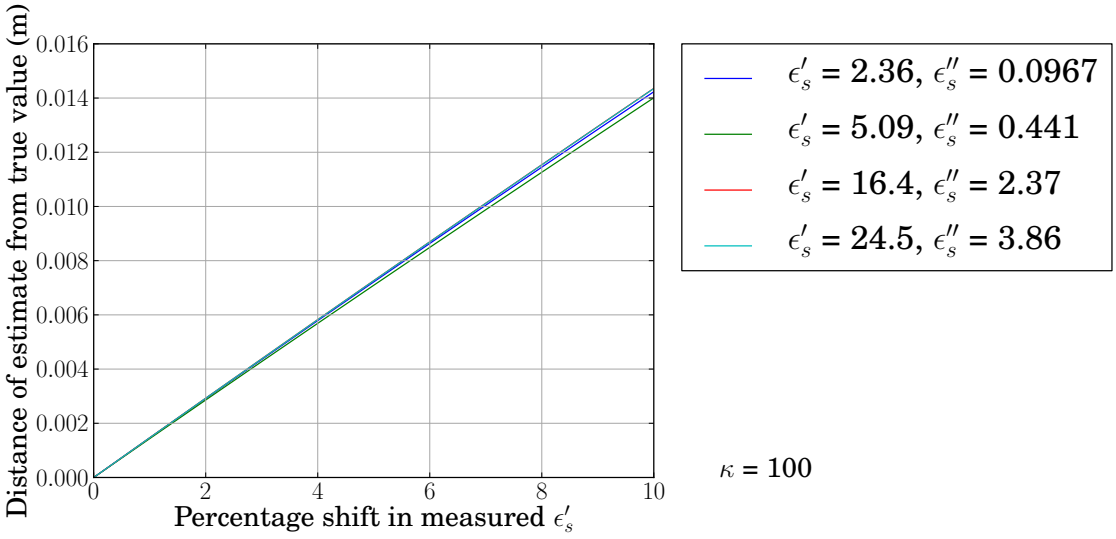


Figure 4.11 Sensitivity analysis of received signal strength based localization w.r.t. ϵ'_s

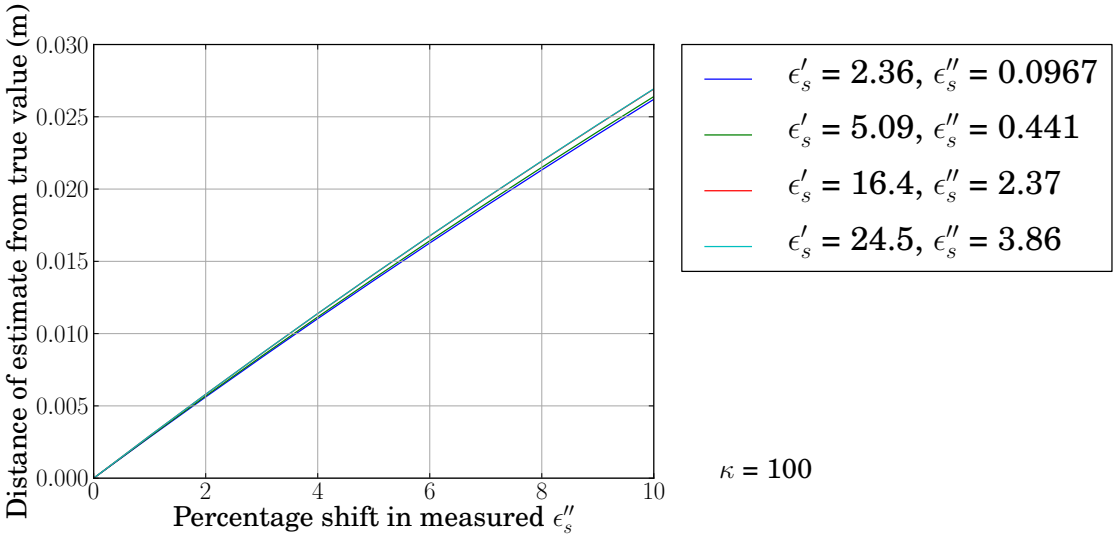


Figure 4.12 Sensitivity analysis of received signal strength based localization w.r.t. ϵ''_s

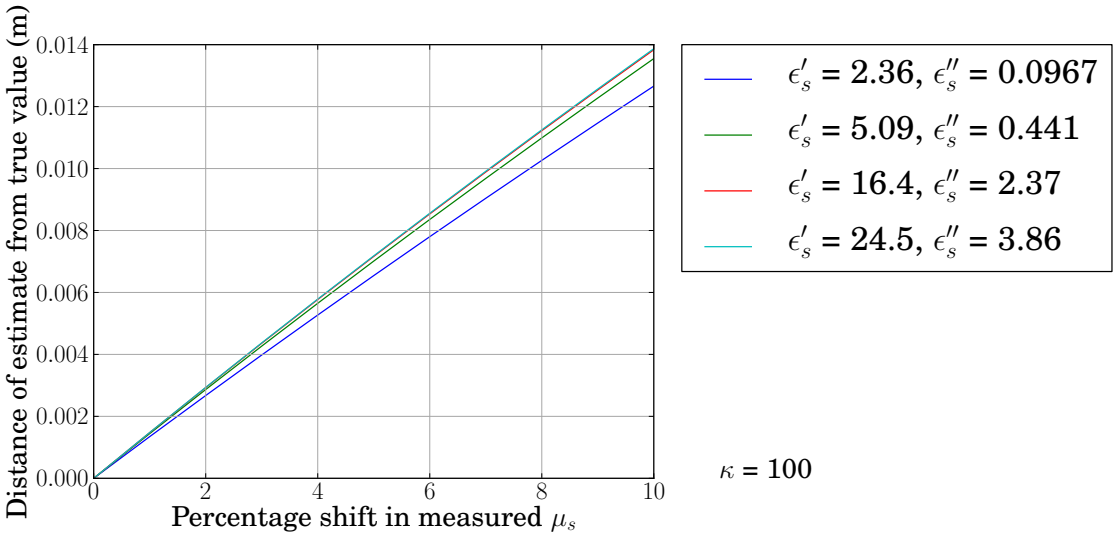


Figure 4.13 Sensitivity analysis of received signal strength based localization w.r.t. μ_s

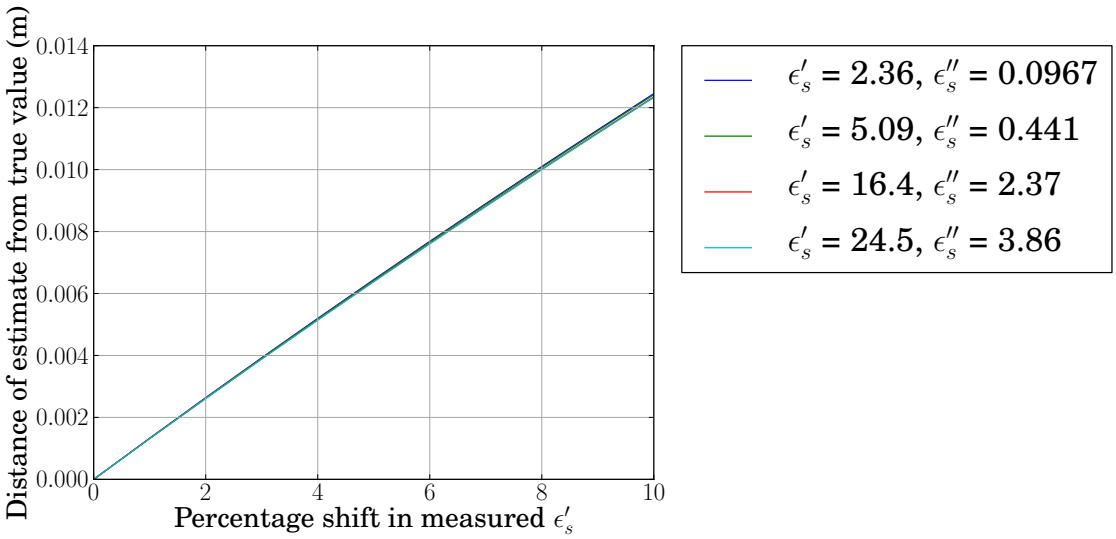


Figure 4.14 Sensitivity analysis of time of arrival based localization w.r.t. ϵ_s'

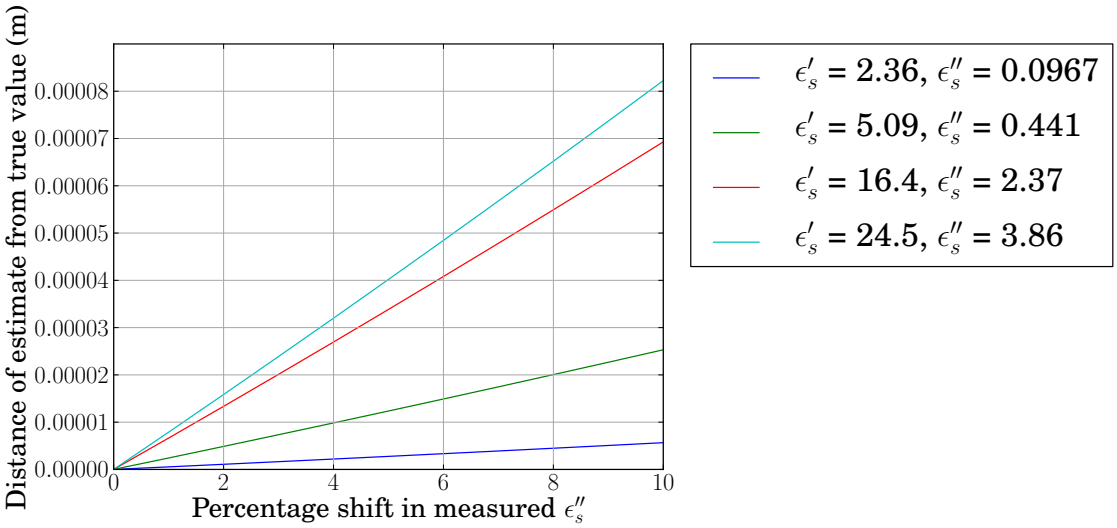


Figure 4.15 Sensitivity analysis of time of arrival based localization w.r.t. ϵ_s''

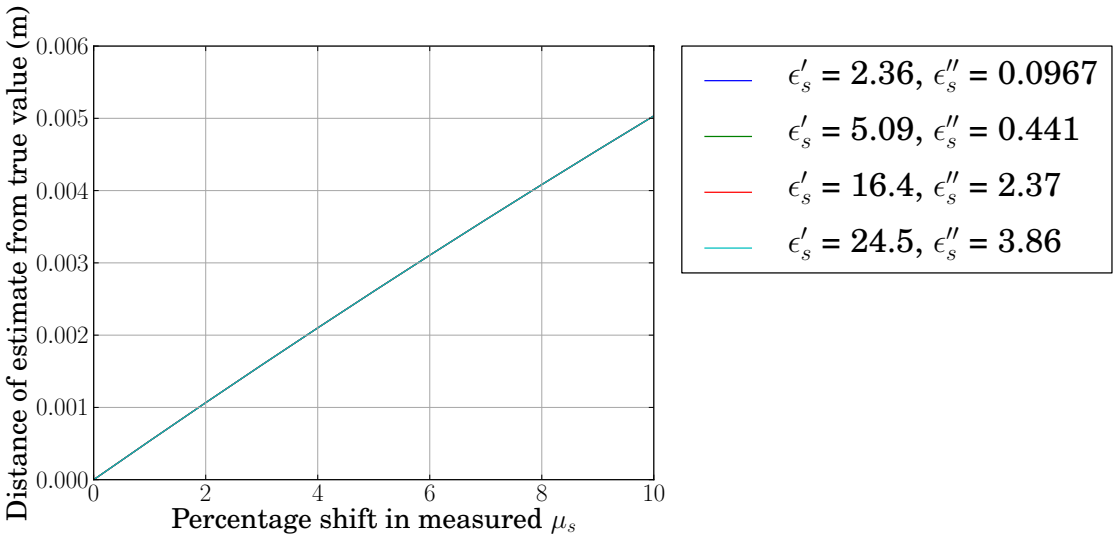


Figure 4.16 Sensitivity analysis of time of arrival based localization w.r.t. μ_s

CHAPTER 5. CONCLUSION AND FUTURE WORK

5.1 Concluding remarks

We study the design of a wireless sensor network system for a precision agriculture application. A number of micro-controller devices integrated with sensors and a radio transceiver are deployed throughout a farm field to sense and collect physical data of interest to various agricultural production systems such as irrigation and fertilizer control. These sensor nodes periodically collect soil data such as temperature, moisture, and fertilizer concentration from spatially distributed locations in the farm field. These devices are battery operated. Thus, their operation must be energy efficient to ensure a long battery life so that the sensor network can collect data over the duration of entire crop season(s) effectively without the need for manual intervention for battery replacements. Our focus is on the energy efficient design of the sensor network system.

Since radio transmission is the most energy consuming operation in a sensor node, an energy efficient network stack provides the highest leverage in maximizing the network lifetime. We design medium access control (MAC) and network protocols for our sensor network in Chapter 2. The sensor nodes periodically transmit the collected data with an interval of an hour or more. Following the traditional approach of powering down the transceiver during the long inactive periods when the nodes are not transmitting, we find that the local clocks of the sensor nodes drift due to the inherent random drifts in the crystal oscillators. Thus, a synchronization mechanism is required for nodes to be able to relay the data to the base station along multiple hops. We employ the approach of trading off the energy consumption at the sender node with that at the receiver node. Furthermore, we exploit the known pattern of data transmissions to minimize the total energy consumption of the system. Our design

proposes a physical (PHY) layer with multiple radio power modes and the upper layers, MAC and network layers, with the goal of reduction in energy consumption, both during wake-up synchronization and data communication. We utilize the event-driven simulation framework of TOSSIM to model the behavior of our PHY/MAC and network layers to compute the energy efficiency of the proposed design. Simulation results show that employing our design can help reduce the energy consumption and also balance the energy consumption profile across all the nodes in a sensor field, reducing the probability of a network partition.

In Chapter 3, we present our MAC layer performance evaluation model to compare the performance of our protocol (PD-MAC) with the traditional duty cycle based S-MAC. The approach is based on a probabilistic analysis to compare metrics such as energy consumption, communication delay and throughput of a network employing PD-MAC with one employing S-MAC, both using the same routing strategy. The results of our model compare well with event driven simulation results. While the two protocols can achieve the same packet success rates, the proposed new protocol is able to accomplish this in 25% less time and 65% less energy (as analyzed and simulated for a simple sensor field of 5×5 nodes). We successfully tested our MAC protocol in hardware employing the new wake-up synchronization strategy for one sender and one receiver case. We also validated our routing and scheduling strategies by deploying a network of nine nodes arranged in a grid fashion that transmit data to a central node over multiple hops.

We address the problem of sensor node localization in Chapter 4. Spatio-temporal data from a widely distributed sensor network in a farm field carries useful meaning for the application only when it is associated with location information. Furthermore, at the network layer, a tree routing strategy such as ours uses the location coordinates of the neighboring nodes to choose the best next hop to route the data towards the sink node. Node localization is a desirable property of a wireless sensor network for our precision agriculture application in which a number of sensor nodes are placed underground while a handful of satellite nodes with known coordinates are located above ground. Location information is also useful for network servicing operations such as battery replacement. We propose methods for three dimensional localization to compute the X, Y coordinates along with the depth Z of the sensor nodes. We

use two types of ranging measurements to estimate the inter-node distances and, consequently, the coordinates of the underground sensor nodes. We use received signal strength and time of arrival of the signal between neighboring underground sensor nodes and between satellite nodes and sensor nodes for estimating the inter-nodal distances.

Received signal strength measurements provide a cost effective ranging solution. However, these measurements are randomly affected by factors such as multi-path fading, reflection, refraction, interference, etc. To the best of our knowledge, past research has not considered localization in a sensor network where nodes are located in different physical media or on modeling the signal strength measurements for transmission across such multiple media. We present a model for the variation in received signal strength due to path loss for transmission in multiple media, multi-path fading, and reflection and refraction across soil-air interface. We use the non-central χ^2 distribution to model the multi-path fading effects for both transmission scenarios: air to soil and soil to soil. The parameters of the distribution are governed by the distance between the receiver and the transmitter nodes, divergence and loss exponents, and Rician factor.

Time of arrival measurements provide a more accurate ranging solution than the received signal strength based approach. We derive a statistical model for the variation of the time of arrival using rigorous analysis. The measured time of arrival is found to be Gaussian distributed with mean governed by the internode distances, and the variance further governed by signal shape, duration, bandwidth and SNR. We pose the localization as a maximum likelihood estimation problem. The improvement in localization accuracy comes at a cost of clock synchronization which, however, is in-built in our set up that requires clocks to be synchronized for communication scheduling.

5.2 Future directions

Wireless sensor networks offer a promising automated data collection solution for precision agriculture but no current solutions are acceptable for deployment at production scales. Researchers are still exploring solutions to the issues that we explored in Section 1.2. An important aspect of such a system is fault tolerance. That is, when a set of nodes in the routing

tree fail, the network must have a way to gracefully recover and minimize the effect on the data collected from the fully functional nodes. We partly address this issue in our network stack by rotating the role of the sink among the four satellite nodes *across* data collection rounds. However, there is potential to further improve the fault tolerance of the system by more adaptive routing techniques *within* data collection rounds. Both detection of faulty nodes and mitigation of the fault's effects are required of a fault management system.

Our localization results for received signal strength are based on the modeling accuracy for multi-path fading and path loss effects. We have modeled the mean signal strength value while keeping the specular to scatter signal power ratio (κ) a constant for a signal propagation path. However, literature results have indicated that the Rician K-factor does change with the propagation distance [63], [64]. Accurately measuring κ is key to localization accuracy during field deployment. The scatter power corresponds to the variance in an AWGN model (used in ToA based localization), discounting the thermal noise. Similarly, a time of arrival based localization model requires parameter calibration using field experiments. Extensive field tests are also needed to validate and calibrate the performance evaluation models.

An improvement in the running time complexity of the localization problem may be achieved by simplifying the optimization problem used for the estimation. For example, formulating the estimation as a method-of-moments problem is expected to speed up the computation of the estimates. However, the improvement in running time may come at the cost of the accuracy of the estimates. An error analysis of the estimates obtained using the method-of-moments approach should be performed.

APPENDIX A. Selected source code - performance evaluation (Objective-C)

Data count computation.

```
void computeP(NSMutableArray* nodes, int forNode, int N)
{
  double Ps = Psuc(N);
  SuccessModel* cNode = [nodes objectAtIndex : forNode];
  //if there are no upstream nodes for a node, then setP
  if([[cNode upstreamNodes] count] == 0)
  {
    [cNode setIsPComputed : YES];
    [cNode setPAtIndex : 1.0 : 1];
  }
  else
  {
    //look at the upstream nodes
    NSArray* upstreamNodesList = [cNode upstreamNodes];
    NSEnumerator* enumerator = [upstreamNodesList objectEnumerator];
    id object;
    BOOL isSet = YES;
    SuccessModel* upstreamNode;
    while((object = [enumerator nextObject]))
    {
      upstreamNode = [nodes objectAtIndex : [object intValue]];
      if([upstreamNode isPComputed] == NO)
      {
        isSet = NO;
      }
    }
    if(isSet == YES)
    {
      enumerator = [upstreamNodesList objectEnumerator];
      NSMutableArray* Pc = [cNode P]; //P0 will be updated here
      //account for dataCount from self
      [cNode setPAtIndex : 1.0 : 1];
      int C = [Pc count];
      PValues* P0 = [[PValues alloc] init];
      [P0 makeP : [cNode P]];
      while((object = [enumerator nextObject]))
      {
        int n, i, r, k;
        upstreamNode = [nodes objectAtIndex : [object intValue]];
        for(n = 1; n <= C;n++)
        {
          double value = [P0 PAtIndex : n]*(1-Ps);
          for(i = 1; i <=  min(n-1, [upstreamNode Pcount]); i++)
```

```
            {
              value = value + [P0 PAtIndex : (n-i)] * [upstreamNode PAtIndex : i]*
                  Ps;
            }
            [cNode setPAtIndex : value : n];
          }
          [P0 release];
          P0 = [[PValues alloc] init];
          [P0 makeP : [cNode P]];
        }
        [cNode setIsPComputed : YES];
      }
    }
    if(forNode == 0)
    {
      FILE* f_out = fopen("/tinyos-2.x/Obj-C/successCountModel.out", "a");
      NSLog(@"Avg success count = %f N = %d", [cNode AvgCount], N);
      fprintf(f_out, "%d \t %f \n",N, [cNode AvgCount]);
      fclose(f_out);
    }
}
int main (int argc, const char * argv[])
{
    NSAutoreleasePool * pool = [[NSAutoreleasePool alloc] init];
    int i, N;
    for(N = 1; N <= 10;N++)
    {
      NSMutableArray* nodeModel = [NSMutableArray arrayWithCapacity : 25];
      for(i = 0; i < 25; i++)
      {
        SuccessModel* model = [[SuccessModel alloc]init];
        [model setUpstreamNodes : i];
        [nodeModel addObject : model];
      }
      for(i = 24; i >= 0; i--)
      {
        //number of hops to sink
        computeP(nodeModel, i, N);
      }

    }
      [pool drain];
      return 0;
}
```

Energy model-PDMAC

```
//@parameters
//N = maximum number of sync opportunities allowed
//n = number of bits of data stored
double Psuc(int N, int dataBits)
{
  double result = 0.0;
  int totalBits = headerBits + dataBits;
  double p = 1 - pow(1-pe,totalBits);
  result = ((1-pow(q,N))*(1-pow(p,L)));
  return result;
```

```
}
//@parameters
//nodes = list of success model for each node
//forNode = node id for which the data count probabilities are being computed
//N = maximum number of synchronization attempts allowed
void computeP(NSMutableArray* nodes, int forNode, int N)
{
  SuccessModel* cNode = [nodes objectAtIndex : forNode];
  //if there are no upstream nodes for a node, then setP
  if([[cNode upstreamNodes] count] == 0)
  {
    [cNode setIsPComputed : YES];
    [cNode setPAtIndex : 1.0 : 1];
  }
  else
  {
    //look at the upstream nodes
    NSArray* upstreamNodesList = [cNode upstreamNodes];
    NSEnumerator* enumerator = [upstreamNodesList objectEnumerator];
    id object;
    BOOL isSet = YES;
    SuccessModel* upstreamNode;
    while((object = [enumerator nextObject]))
    {
      upstreamNode = [nodes objectAtIndex : [object intValue]];
      if([upstreamNode isPComputed] == NO)
      {
        isSet = NO;
      }
    }
    if(isSet == YES)
    {
      enumerator = [upstreamNodesList objectEnumerator];
      NSMutableArray* Pc = [cNode P]; //P0 will be updated here
      //account for dataCount from self
      [cNode setPAtIndex : 1.0 : 1];
      int C = [Pc count];
      PValues* P0 = [[PValues alloc] init];
      [P0 makeP : [cNode P]];
      while((object = [enumerator nextObject]))
      {
        int n;
        int i;
        upstreamNode = [nodes objectAtIndex : [object intValue]];
        //all fail loop, replacing already existent values in P
        for(n = 1; n <= C; n++)
        {
          //each data unit is one byte
          //all fail case
          double PFail = 0.0;
          for (int j = 1; j <=  [upstreamNode Pcount]; j++)
          {
            PFail = PFail + [upstreamNode PAtIndex : j]*(1 - Psuc(N, j*dataBits))
                ;
          }
          double value = [P0 PAtIndex : n] * (PFail);
          for(i = 1; i <=  min(n-1, [upstreamNode Pcount]); i++)
          {
            value = value + [P0 PAtIndex : (n-i)] * [upstreamNode PAtIndex : i] *
                Psuc(N, i * 8);
```

```
            }
            [cNode setPAtIndex : value : n];
          }
          [P0 release];
          P0 = [[PValues alloc] init];
          [P0 makeP : [cNode P]];
        }
        [cNode setIsPComputed : YES];
      }
    }
}
int main (int argc, const char * argv[])
{
  NSAutoreleasePool * pool = [[NSAutoreleasePool alloc] init];
  FILE* ofile = fopen("/tinyos-2.x/Obj-C/EnergyPdmac/EnergyModel.out", "w");
  fprintf(ofile, "PDMAC \n q = %f, pe = %f, headerBits = %d, dataBits = %d, Ping
      duration = %f\n",q, pe, headerBits, dataBits, TP);
  int i, N;
  double energy;
  for(N = 1;N <= 10;N++)
  {
    //create the probability distribution of the number of data readings stored
        at each node
    NSMutableArray* nodeSuccessModel = [NSMutableArray arrayWithCapacity : 25];
    NSMutableArray* nodeEnergyModel = [NSMutableArray arrayWithCapacity : 25];
    for(i = 0; i < M; i++)
    {
      SuccessModel* model = [[SuccessModel alloc]init];
      [model setUpstreamNodes : i];
      [nodeSuccessModel addObject : model];
    }
    //compute the probability distribution of dataCount at each node
    for(i = M-1;i>= 0;i--)
    {
      computeP(nodeSuccessModel, i, N);
    }
    //Print the successModel for node 0 here
    //NSLog(@"Avg Count for N = %d = %f, ", N, [[nodeSuccessModel objectAtIndex :
        0] AvgCount]);
    energy = 0.0;
    for(i = 0; i < M; i++)
    {
      //set the current nodeId and the downstream nodeId
      [nodeEnergyModel addObject : [[Node alloc] initForId : i]];
      //this will give the list of all the upstream nodeIds in an array of
          NSNumbers intvalues
      NSArray *upstreamNodesList = [[nodeSuccessModel objectAtIndex : i]
          upstreamNodes];
      for(id uNode in upstreamNodesList)
      {
      //adding successmodel for all upstream nodes of node i
        [[nodeEnergyModel objectAtIndex : i] addUpstreamNode : [nodeSuccessModel
            objectAtIndex : [uNode intValue]]];   //SuccessModel for upstream
            node
      }
    }
    //now set the success models for the peers, node 0 does not have any peers
    //look at the downstream node id of current node, and get the successmodels
        for all the upstream neighbors
    //of that node
```

```
    //set the peernodes successmodel for the current node to the just returned
       upstream nodes successmodel
    //will have to search in that array to find the entry for the current node
    for(int i = 1; i < M; i++)
    {
      int dNode = [[nodeEnergyModel objectAtIndex : i] downstreamNode];
      NSMutableArray* peerNodesSuccessModel = [NSMutableArray arrayWithArray : [[
          nodeEnergyModel objectAtIndex : dNode] upstreamNodes]];
      int pNodeIndex = 0;
      for(id pNode in peerNodesSuccessModel)
      {
        if(([pNode nodeId] == i)&&(pNodeIndex! = 0))
        {
          [peerNodesSuccessModel exchangeObjectAtIndex : 0 withObjectAtIndex :
              pNodeIndex];
        }
        pNodeIndex++;
      }
      [[nodeEnergyModel objectAtIndex : i] addPeerNodeModel :
          peerNodesSuccessModel];
      //Now compute the energy for all nodes, for debugging I do it only for node
          1 now
      //sum up the energy for all nodes here for num opps = N
            energy = energy + [[nodeEnergyModel objectAtIndex : i] energy];
    }
    for(int i = 0; i < M; i++)
    {
      [[nodeEnergyModel objectAtIndex : i] computeEnergyForPingOpps : N];
      energy = energy + [[nodeEnergyModel objectAtIndex : i] energy];
    }
    NSLog(@"Avg energy for num opps = %d = %f", N, energy/M);
  }
  fclose(ofile);
    [pool drain];
    return 0;
}
```

Energy model-SMAC

```
//@parameters
//N = maximum number of sync opportunities allowed
//n = number of bits of data stored
double Psuc(int N, int dataBits)
{
  double result = 0.0;
  int totalBits = headerBits + dataBits;
  double p = 1 - pow(1-pe,totalBits);
  double q = 1-pow(1-pe, headerBits+syncBits);
  result = ((1-pow(q,N))*(1-pow(p,L)));
  return result;
}
int min(int a, int b)
{
  if(a < b)
    return a;
  else
    return b;
```

```
}
//@parameters
//nodes = list of success model for each node
//forNode = node id for which the data count probabilities are being computed
//N = maximum number of synchronization attempts allowed
void computeP(NSMutableArray* nodes, int forNode, int N)
{
  SuccessModel* cNode = [nodes objectAtIndex : forNode];
  //if there are no upstream nodes for a node, then setP
  if([[cNode upstreamNodes] count] == 0)
  {
    [cNode setIsPComputed : YES];
    [cNode setPAtIndex : 1.0 : 1];
    return;
  }
  else
  {
    //look at the upstream nodes
    NSArray* upstreamNodesList = [cNode upstreamNodes];
    BOOL isSet = YES;
    for(id object in upstreamNodesList)
    {
      SuccessModel* upstreamNode = [nodes objectAtIndex : [object intValue]];
      if([upstreamNode isPComputed] == NO)
      {
        NSLog(@"P for up nodes is not computed, check!!!");
        isSet = NO;
        break;
      }
    }
    if(isSet == YES)
    {
      NSMutableArray* Pc = [cNode P]; //P0 will be updated here
      [cNode setPAtIndex : 1.0 : 1];
      int C = [Pc count];
      PValues* P0 = [[PValues alloc] init];
      [P0 makeP : [cNode P]];
      for(id object in upstreamNodesList)
      {
        int n, i;
        SuccessModel* upstreamNode = [nodes objectAtIndex : [object intValue]];
        for(n = 1; n <= C; n++)
        {
          //each data unit is one byte
          double PFail = 0.0;
          for (int j = 1; j <=  [upstreamNode Pcount]; j++)
          {
            PFail = PFail + [upstreamNode PAtIndex : j]*(1-Psuc(N, j*dataBits));
          }
          //all fail from upnode
          double value = [P0 PAtIndex : n] * (PFail);
          for(i = 1; i <=  min(n-1, [upstreamNode Pcount]); i++)
          {
            value = value + [P0 PAtIndex : (n-i)] * [upstreamNode PAtIndex : i] *
                Psuc(N, i * dataBits);
          }
          [cNode setPAtIndex : value : n];
        }
        [P0 release];
        P0 = [[PValues alloc] init];
```

```
        [P0 makeP : [cNode P]];
      }
      [cNode setIsPComputed : YES];
    }
    return;
  }
}
int main (int argc, const char * argv[])
{
   NSAutoreleasePool * pool = [[NSAutoreleasePool alloc] init];
  for(int N = 1; N <= 10; N++)
  {
    //create the probability distribution of the number of
    //data readings stored at each node
    NSMutableArray* nodeSuccessModel = [NSMutableArray arrayWithCapacity : 25];
    NSMutableArray* nodeEnergyModel = [NSMutableArray arrayWithCapacity : 25];
    for(int i = 0; i < M; i++)
    {
      SuccessModel* model = [[SuccessModel alloc] init];
      [model setUpstreamNodes : i];
      [nodeSuccessModel addObject : model];
    }
    for(int i = M - 1; i >= 0; i--)
    {
      computeP(nodeSuccessModel, i, N);
    }
    double energy = 0.0;
    //add the success models for the upstream nodes for all the nodes
    for(int i  = 0; i < M; i++)
    {
      [nodeEnergyModel addObject : [[Node alloc] initForId : i]];
      NSArray* upstreamNodesList = [[nodeSuccessModel objectAtIndex : i]
          upstreamNodes];
      for(id uNode in upstreamNodesList)
      {
        [[nodeEnergyModel objectAtIndex : i] addUpstreamNode : [nodeSuccessModel
            objectAtIndex : [uNode intValue]]];
      }
    }
    //add success model for self
    for(int i = 0; i < M; i++)
    {
      [[nodeEnergyModel objectAtIndex : i] addSelfSuccessModel : [
          nodeSuccessModel objectAtIndex : i]];
    }
    //compute the energy consumption for all nodes
    for(int i = 0; i < M; i++)
    {
      [[nodeEnergyModel objectAtIndex : i] computeEnergyForPingOpps : N];
      energy = energy + [[nodeEnergyModel objectAtIndex : i] energy];
    }
    NSLog(@"N = %d, energy = %f", N, energy/25);
  }
    [pool drain];
    return 0;
}
```

Duration model-PDMAC

```
//
//  ModifiedDurationModel.m
//  PDMACDelayModel
//
//  Created by Herman Sahota on 10/19/10.
//  Copyright 2010 Iowa State University. All rights reserved.
//
@implementation DurationModel
-(id) init
{
  if(self = [super init])
  {
  }
  return(self);
}
-(float) computeDuration : (int) N
{
  float result = 0.0;
  double dataTime = 0.0;
  for(id upNode in senders)
  {
    int maxBits = [upNode Pcount]*dataBits + headerBits;
    dataTime = dataTime + maxBits/BPS;
  }
  double ackTime = (headerBits+numS)/BPS;
  //total time the downstream node is awake including the starting of ping
  double dNodeTimeAwake = [AverageFunctions downstreamNodeAwakeTime : senders : 0
       : N : dataTime : ackTime];
  //add to this the longest duration of time a sender node spends in drowsy (
      excluding reception of ping)
  result = 3 * delta + dNodeTimeAwake;
  return result;
}
-(id) initForRx : (int) i
{
  if(self = [super init])
  {
    rxId = i;
    delay = 0.0;
    senders = [NSMutableArray arrayWithCapacity : 3];
  }
  return self;
}
-(void) addSender : (SuccessModel* ) sender
{
  [senders addObject : sender];
  numS = [senders count];
}
-(int) rxId
{
  return rxId;
}
@end
```

APPENDIX B. Selected code - Sensor node localization (Python)

Data generation and likelihood functions for received signal strength based localization

```
#AIR TO SOIL DATA GENERATION
def d_as_s(m1, m2, n):
    return ZActual[n]

def d_as_a(m1, m2, n):
    return sqrt( (XSatellite[m1][m2]-XActual[n]) ** 2 + (YSatellite[m1][m2]-
        YActual[n]) ** 2 + (ZSatellite[m1][m2]) ** 2)

def p_average_air2soil(d_a, d_s, ALPHA_SOIL, TAU):
  p = PA_0 * (LAMBDA ** K_SOIL) / ((4 * pi * (d_s + d_a)) ** K_AIR) *\
                exp(-2 * ALPHA_SOIL * d_s) * TAU
  return p


def mean_p_as(m1, m2, n, ALPHA_SOIL, TAU):
  d_a = d_as_a(m1, m2, n)
  d_s = d_as_s(m1, m2, n)
  return p_average_air2soil(d_a, d_s, ALPHA_SOIL, TAU)

def sample_rss_as(m1, m2, n, ALPHA_SOIL, TAU):
    pmn = mean_p_as(m1, m2, n, ALPHA_SOIL, TAU)
    sigma2 = pmn/(2 * (KAPPA + 1))
    s = ncx2.rvs(2, 2 * KAPPA, size = 1, scale = sigma2)
    return s[0]


#SOIL TO SOIL DATA GENERATION

def d_ss_dir(m, n):
  return sqrt( (XActual[m] - XActual[n]) ** 2  +\
     (YActual[m] - YActual[n]) ** 2  + \
    (ZActual[m] - ZActual[n]) ** 2  )

def d_ss_ref(m, n):
  return sqrt( ( pow(XActual[m] - XActual[n], 2) + pow(YActual[m] - YActual[n],2)
       ) * pow(ZActual[n],2) / pow(ZActual[m] + ZActual[n], 2) + pow(ZActual[n],
      2)) \
  + sqrt( (pow(XActual[m]-XActual[n], 2) + pow(YActual[m] - YActual[n], 2)) * pow
      (ZActual[m], 2) / pow(ZActual[m] + ZActual[n], 2) + pow(ZActual[m], 2) )
```

```
def p_average_soil2soil_total(d_dir, d_ref, ALPHA_SOIL, RHO):
  direct_power = PS_0 * (LAMBDA ** K_SOIL) / ((4 * pi * d_dir) ** K_SOIL) *\
                 exp(-2 * ALPHA_SOIL * d_dir)
  reflected_power = PS_0 * (LAMBDA ** K_SOIL) / ((4 * pi * d_ref) ** K_SOIL) *\
                 exp(-2 * ALPHA_SOIL * d_ref) * RHO
  return  direct_power + reflected_power

def sample_rss_ss(m, n, ALPHA_SOIL, RHO):

    d_dir = d_ss_dir(m, n)
    d_ref = d_ss_ref(m, n)
    pmn = p_average_soil2soil_total(d_dir, d_ref, ALPHA_SOIL, RHO)
    sigma2 = pmn / (2 * (KAPPA + 1));
    s = ncx2.rvs(2, 2 * KAPPA, size = 1, scale = sigma2)
    return s[0]


# Data generation
def generate_data_as(thisNodeId, ALPHA_SOIL, TAU):
  observed_rss_as = []
  for m1 in range(0, 2):
    for m2 in range(0, 2):
      for i in range(0, NUM_READINGS_PER_NODE_PAIR_AS):
        observed_rss_as_ = sample_rss_as(m1, m2, thisNodeId, ALPHA_SOIL, TAU)
        if(10 * log(observed_rss_as_, 10) > -110.0):
          observed_rss_as.append([m1, m2, thisNodeId, observed_rss_as_])
  return {"data": observed_rss_as}

def generate_data_ss(thisNodeId, ALPHA_SOIL, RHO):
  observed_rss_ss = []
  neighbors = []
  for m in range(NUM_SENSORS):
    if(m==thisNodeId):
      continue
    neighbor_appended = False
    for i in range(0, NUM_READINGS_PER_NODE_PAIR_SS):
      observed_rss_ss_ = sample_rss_ss(m, thisNodeId, ALPHA_SOIL, RHO)

      if observed_rss_ss_ > 0.0 and 10 * log(observed_rss_ss_, 10) > -110.0:
        observed_rss_ss.append([m, thisNodeId, observed_rss_ss_])
        if(neighbor_appended == False):
          neighbor_appended = True
          neighbors.append(m)
  return {"data": observed_rss_ss, "neighbors": neighbors}

# LIKELIHOOD FUNCTIONS

def negative_log_likelihood_xyz_as(arg, n, observed_rss_as, ALPHA_SOIL, TAU):

  x = arg[0]
  y = arg[1]
  z = arg[2] * MAX_DEPTH / FIELD_LENGTH


  likelihood = 0.0

  rss_obs = {'11':[], '01': [], '10': [], '00': []}
  rss_est = {}
```

```
  for m1 in range(0, 2):
    for m2 in range(0, 2):
      d_s_est = z

      d_a_est = np.sqrt(\
          (XSatellite[m1][m2]-x) ** 2 +\
         (YSatellite[m1][m2]-y) ** 2 +\
          (ZSatellite[m1][m2]) ** 2\
          )

      rss_est_ = p_average_air2soil(d_a_est, d_s_est, ALPHA_SOIL, TAU)
      rss_est[str(m1)+str(m2)] = rss_est_

  for data in observed_rss_as:
    m1 = data[0]
    m2 = data[1]
    rss_obs[str(m1)+str(m2)].append(data[3])

  for m1 in range(0, 2):
    for m2 in range(0, 2):
      d_s_est = z

      d_a_est = np.sqrt(\
          (XSatellite[m1][m2]-x) ** 2 +\
         (YSatellite[m1][m2]-y) ** 2 +\
          (ZSatellite[m1][m2]) ** 2\
          )
      pmn = p_average_air2soil(d_a_est, d_s_est, ALPHA_SOIL, TAU)

      sigma2_est = pmn/(2 * (KAPPA + 1))
      if(math.isnan(sigma2_est)):
        return 9999.9
      for Rmn in rss_obs[str(m1)+str(m2)]:
        if(Rmn/sigma2_est > 1000):
          return 9999.0
        likelihood  = likelihood - ncx2.logpdf(Rmn/sigma2_est, 2, 2 * KAPPA)
  return likelihood


def negative_log_likelihood_xyz_ss(arg, thisNodeId, curr_estimates,
    observed_rss_ss, neighbors, ALPHA_SOIL, RHO):
  x = arg[0]
  y = arg[1]

  XEstimates = curr_estimates["x"]
  YEstimates = curr_estimates["y"]
  ZEstimates = curr_estimates["z"]

  likelihood = 0.0
  n = thisNodeId
  if len(neighbors) < 2:
    return 0.0

  for m in range(NUM_SENSORS):
    if not m in neighbors:
      continue

    d_dir_est = sqrt( (XEstimates[m] - x) ** 2  + \
       (YEstimates[m] - y) ** 2  + \
```

```
      (ZEstimates[m] - ZActual[n]) ** 2  )

    d_ref_est = sqrt( ( pow(XEstimates[m] - x, 2) + pow(YEstimates[m] - y,2) ) *
        pow(ZActual[n],2) / pow(ZEstimates[m] + ZActual[n], 2) + pow(ZActual[n],
        2)) \
  + sqrt( (pow(XEstimates[m] - x, 2) + pow(YEstimates[m] - y, 2)) * pow(
      ZEstimates[m], 2) / pow(ZEstimates[m] + ZActual[n], 2) + pow(ZEstimates[m],
      2) )

    pmn = p_average_soil2soil_total(d_dir_est, d_ref_est, ALPHA_SOIL, RHO)
    sigma2_est = pmn/(2 * (KAPPA + 1))

    rss_obs = []
    for data in observed_rss_ss:
      if not (data[0] == m):
        continue
      rss_obs.append(data[2])

    for Rmn in rss_obs:
      likelihood  = likelihood - ncx2.logpdf(Rmn/sigma2_est, 2, 2 * KAPPA)
    print likelihood
  return likelihood


def negative_log_likelihood(arg, thisNodeId, curr_estimates, observed_rss_ss,
    observed_rss_as, neighbors, ALPHA_SOIL, RHO, TAU):

  ss_log_likelihood = negative_log_likelihood_xyz_ss([arg[0], arg[1]], thisNodeId
      , curr_estimates, observed_rss_ss, neighbors, ALPHA_SOIL, RHO)
  as_log_likelihood = 10 * negative_log_likelihood_xyz_as(arg, thisNodeId,
      observed_rss_as, ALPHA_SOIL, TAU)

  return as_log_likelihood + ss_log_likelihood
```

BIBLIOGRAPHY

[1] Junyan Ma, Xingshe Zhou, Shining Li, and Zhigang Li. Connecting agriculture to the internet of things through sensor networks. In *Internet of Things (iThings/CPSCom), 2011 International Conference on and 4th International Conference on Cyber, Physical and Social Computing*, pages 184 –187, oct. 2011.

[2] J. Burrel, T. Brooke, and R. Beckwith. Vineyard computing: sensor networks in agricultural production. *Pervasive Computing, IEEE*, 3(1):38–45, jan-march 2004.

[3] R. Beckwith, D. Teibel, and P. Bowen. Unwired wine: sensor networks in vineyards. In *Sensors, 2004. Proceedings of IEEE*, pages 561 – 564 vol.2, oct. 2004.

[4] Jianfa Xia, Zhenzhou Tang, Xiaoqiu Shi, Lei Fan, and Huaizhong Li. An environment monitoring system for precise agriculture based on wireless sensor networks. In *Mobile Ad-hoc and Sensor Networks (MSN), 2011 Seventh International Conference on*, pages 28 –35, dec. 2011.

[5] Jain Cai and D. Goodman. General packet radio service in gsm. *Communications Magazine, IEEE*, 35(10):122–131, 1997.

[6] M. Purvis, J. Sambells, and C. Turner. *Beginning Google maps applications with PHP and Ajax*. Michael Purvis, Jeffrey Sambells, and Cameron Turner, 2006.

[7] J. Polastre, R. Szewczyk, and D. Culler. Telos: enabling ultra-low power wireless research. In *Information Processing in Sensor Networks, 2005. IPSN 2005. Fourth International Symposium on*, pages 364 – 369, april 2005.

[8] Wei Ye, John Heidemann, and Deborah Estrin. Medium access control with coordinated adaptive sleeping for wireless sensor networks. *IEEE/ACM Trans. Netw.*, 12(3):493–506, 2004.

[9] Miklós Maróti, Branislav Kusy, Gyula Simon, and Ákos Lédeczi. The flooding time synchronization protocol. In *Proceedings of the 2nd international conference on Embedded networked sensor systems*, SenSys '04, pages 39–49, New York, NY, USA, 2004. ACM.

[10] S. Verma, N. Chug, and D.V. Gadre. Wireless sensor network for crop field monitoring. In *Recent Trends in Information, Telecommunication and Computing (ITC), 2010 International Conference on*, pages 207 –211, march 2010.

[11] Kyeong og Kim, Jong chan Kim, Kyeong jin Ban, Eung kon Kim, and Moon suk Jang. U-it based greenhouse environment monitoring system. In *Multimedia and Ubiquitous Engineering (MUE), 2011 5th FTRA International Conference on*, pages 203 –206, june 2011.

[12] J. Myster and R. Moe. Effect of diurnal temperature alternations on plant morphology in some greenhouse crops—a mini review. *Scientia Horticulturae*, 62(4):205–215, 1995.

[13] L. H. M. Cuijpers and J. V. M. Vogelezang. DIF and temperature drop for short-day pot plants. *Acta Hort. (ISHS)*, 327:25–32, 1992.

[14] Yuan Yuan, Shanshan Li, Kui Wu, Weijia Jia, and Yuxing Peng. Focus: A cost-effective approach for large-scale crop monitoring with sensor networks. In *Mobile Adhoc and Sensor Systems, 2009. MASS '09. IEEE 6th International Conference on*, pages 544 –553, oct. 2009.

[15] B.G. Jagyasi, A.K. Pande, and R. Jain. Event based experiential computing in agro-advisory system for rural farmers. In *Wireless and Mobile Computing, Networking and Communications (WiMob), 2011 IEEE 7th International Conference on*, pages 439 –444, oct. 2011.

[16] Guoming Sang and Liwei Song. The design and implementation of a farmland monitoring wireless sensor network. In *Circuits,Communications and System (PACCS), 2010 Second Pacific-Asia Conference on*, volume 1, pages 355 –358, aug. 2010.

[17] W. Chebbi, M. Benjemaa, A. Kamoun, M. Jabloun, and A. Sahli. Development of a wsn integrated weather station node for an irrigation alert program under tunisian conditions. In *Systems, Signals and Devices (SSD), 2011 8th International Multi-Conference on*, pages 1 –6, march 2011.

[18] Yong Huang. Design and realization of wireless sensor network for vegetable greenhouse information acquisition. In *Wireless Communications Networking and Mobile Computing (WiCOM), 2010 6th International Conference on*, pages 1 –4, sept. 2010.

[19] Yunseop Kim, R.G. Evans, and W.M. Iversen. Remote sensing and control of an irrigation system using a distributed wireless sensor network. *Instrumentation and Measurement, IEEE Transactions on*, 57(7):1379 –1387, july 2008.

[20] K. Langendoen, A. Baggio, and O. Visser. Murphy loves potatoes: experiences from a pilot sensor network deployment in precision agriculture. In *Parallel and Distributed Processing Symposium, 2006. IPDPS 2006. 20th International*, page 8 pp., april 2006.

[21] H. Sahota, R. Kumar, A. Kamal, and J. Huang. An energy-efficient wireless sensor network for precision agriculture. In *Proc. IEEE Symposium on Computers and Communications*, pages 347–350, Riccione, Italy, June 2010. IEEE Computer Society.

[22] Philip Levis, Nelson Lee, Matt Welsh, and David Culler. Tossim: accurate and scalable simulation of entire tinyos applications. In *SenSys '03: Proceedings of the 1st international conference on Embedded networked sensor systems*, pages 126–137, New York, NY, USA, 2003. ACM.

[23] Texas Instruments. True system-on-chip with low power rf transceiver and 8051 mcu. http://www.ti.com/lit/gpn/cc1110f32.

[24] Tijs van Dam and Koen Langendoen. An adaptive energy-efficient mac protocol for wireless sensor networks. In *SenSys '03: Proceedings of the 1st international conference on Embedded networked sensor systems*, pages 171–180, New York, NY, USA, 2003. ACM.

[25] P. Lin, C. Qiao, and X. Wang. Medium access control with a dynamic duty cycle for sensor networks. In *IEEE Wireless Communications and Networking Conference*, volume 3, pages 1534 – 1539, March 2004.

[26] Joseph Polastre, Jason Hill, and David Culler. Versatile low power media access for wireless sensor networks. In *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 95–107, New York, NY, USA, 2004. ACM.

[27] A. El-Hoiydi and J.-D. Decotignie. Wisemac: an ultra low power mac protocol for the downlink of infrastructure wireless sensor networks. In *ISCC '04: Proceedings of the Ninth International Symposium on Computers and Communications 2004 Volume 2 (ISCC"04)*, pages 244–251, Washington, DC, USA, 2004. IEEE Computer Society.

[28] Antonio G. Ruzzelli, Raja Jurdak, and Gregory M.P. O'Hare. On the rfid wake-up impulse for multi-hop sensor networks. In *1st ACM Workshop on Convergence of RFID and Wireless Sensor Networks and their Applications (SenseID) at the Fifth ACM Conference on Embedded Networked Sensor Systems (ACM SenSys 2007)*. ACM, November 2007.

[29] The TinyOS 2.x Working Group. Tinyos 2.0. In *SenSys '05: Proceedings of the 3rd international conference on Embedded networked sensor systems*, pages 320–320, New York, NY, USA, 2005. ACM.

[30] Guido Rossum. Python reference manual, 1995.

[31] Vijay Erramilli, Ibrahim Matta, and Azer Bestavros. On the interaction between data aggregation and topology control in wireless sensor networks, 2004.

[32] C. F Chiasserini and M. Garetto. Modeling the performance of wireless sensor networks. In *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, volume 1, pages –231, 2004.

[33] M. Noori and M. Ardakani. Characterizing the traffic distribution in linear wireless sensor networks. *Communications Letters, IEEE*, 12(8):554–556, 2008.

[34] Yong He, Ruixi Yuan, and W. Gong. Modeling power saving protocols for multicast services in 802.11 wireless lans. *Mobile Computing, IEEE Transactions on*, 9(5):657–671, 2010.

[35] K. Ramachandran and B. Sikdar. A population based approach to model the lifetime and energy distribution in battery constrained wireless sensor networks. *Selected Areas in Communications, IEEE Journal on*, 28(4):576–586, 2010.

[36] H. Husni, N.I.M. Enzai, N.A.M. Rais, and Z. Jusoh. Evaluation of random node shutdown in wireless sensor network for improving energy efficiency. In *Business Engineering and Industrial Applications Colloquium (BEIAC), 2012 IEEE*, pages 64–69, 2012.

[37] Guihai Chen, Chengfa Li, Mao Ye, and Jie Wu. An unequal cluster-based routing protocol in wireless sensor networks. *Wirel. Netw.*, 15(2):193–207, February 2009.

[38] D. Hasenfratz, A. Meier, C. Moser, Jian-Jia Chen, and L. Thiele. Analysis, comparison, and optimization of routing protocols for energy harvesting wireless sensor networks. In *Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC), 2010 IEEE International Conference on*, pages 19–26, 2010.

[39] A.D.G. Dimakis, A.D. Sarwate, and M.J. Wainwright. Geographic gossip: Efficient averaging for sensor networks. *Signal Processing, IEEE Transactions on*, 56(3):1205 –1216, march 2008.

[40] Kai Zeng, Kui Ren, Wenjing Lou, and Patrick J. Moran. Energy aware efficient geographic routing in lossy wireless sensor networks with environmental energy supply. *Wirel. Netw.*, 15(1):39–51, January 2009.

[41] Dimitrios Koutsonikolas, Saumitra M. Das, Y. Charlie Hu, and Ivan Stojmenovic. Hierarchical geographic multicast routing for wireless sensor networks. *Wirel. Netw.*, 16(2):449–466, February 2010.

[42] Marco Zúñiga Zamalloa, Karim Seada, Bhaskar Krishnamachari, and Ahmed Helmy. Efficient geographic routing over lossy links in wireless sensor networks. *ACM Trans. Sen. Netw.*, 4(3):12:1–12:33, June 2008.

[43] Kui Ren, Wenjing Lou, and Yanchao Zhang. Leds: Providing location-aware end-to-end data security in wireless sensor networks. *Mobile Computing, IEEE Transactions on*, 7(5):585 –598, may 2008.

[44] N. Patwari, J.N. Ash, S. Kyperountas, III Hero, A.O., R.L. Moses, and N.S. Correal. Locating the nodes: cooperative localization in wireless sensor networks. *Signal Processing Magazine, IEEE*, 22(4):54 – 69, july 2005.

[45] J. Aspnes, T. Eren, D.K. Goldenberg, A.S. Morse, W. Whiteley, Y.R. Yang, B.D.O. Anderson, and P.N. Belhumeur. A theory of network localization. *Mobile Computing, IEEE Transactions on*, 5(12):1663 –1678, dec. 2006.

[46] Nissanka B. Priyantha, Hari Balakrishnan, Erik Demaine, and Seth Teller. Poster abstract: anchor-free distributed localization in sensor networks. In *Proceedings of the 1st international conference on Embedded networked sensor systems*, SenSys '03, pages 340–341, New York, NY, USA, 2003. ACM.

[47] A. Youssef, A. Agrawala, and M. Younis. Accurate anchor-free node localization in wireless sensor networks. In *Performance, Computing, and Communications Conference, 2005. IPCCC 2005. 24th IEEE International*, pages 465–470, 2005.

[48] N. Bulusu, J. Heidemann, and D. Estrin. Gps-less low-cost outdoor localization for very small devices. *Personal Communications, IEEE*, 7(5):28–34, 2000.

[49] L. Doherty, K.S.J. pister, and L. El Ghaoui. Convex position estimation in wireless sensor networks. In *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 3, pages 1655–1663 vol.3, 2001.

[50] C. Savarese, J.M. Rabaey, and J. Beutel. Location in distributed ad-hoc wireless sensor

networks. In *Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP '01). 2001 IEEE International Conference on*, volume 4, pages 2037–2040 vol.4, 2001.

[51] Andreas Savvides, Chih-Chieh Han, and Mani B. Strivastava. Dynamic fine-grained localization in ad-hoc networks of sensors. In *Proceedings of the 7th annual international conference on Mobile computing and networking*, MobiCom '01, pages 166–179, New York, NY, USA, 2001. ACM.

[52] Bin Xiao, Lin Chen, Qingjun Xiao, and Minglu Li. Reliable anchor-based sensor localization in irregular areas. *Mobile Computing, IEEE Transactions on*, 9(1):60–72, 2010.

[53] Y. Shang, W. Rumi, Y. Zhang, and M. Fromherz. Localization from connectivity in sensor networks. *Parallel and Distributed Systems, IEEE Transactions on*, 15(11):961 – 974, nov. 2004.

[54] I. Borg and P.J.F. Groenen. *Modern multidimensional scaling: theory and applications : with 116 figures*. Springer Series in Statistics Series. Springer-Verlag GmbH, 1997.

[55] R. Stoleru, Tian He, S.S. Mathiharan, S.M. George, and J.A. Stankovic. Asymmetric event-driven node localization in wireless sensor networks. *Parallel and Distributed Systems, IEEE Transactions on*, 23(4):634–642, 2012.

[56] Ziguo Zhong and Tian He. Sensor node localization with uncontrolled events. *ACM Trans. Embed. Comput. Syst.*, 11(3):65:1–65:25, September 2012.

[57] H. Wymeersch, J. Lien, and M.Z. Win. Cooperative localization in wireless networks. *Proceedings of the IEEE*, 97(2):427 –450, feb. 2009.

[58] Bram Dil, Stefan Dulman, and Paul Havinga. Range-based localization in mobile sensor networks. In *Wireless Sensor Networks*, pages 164–179. Springer, 2006.

[59] A. Savvides, W.L. Garber, R.L. Moses, and M.B. Srivastava. An analysis of error inducing parameters in multihop sensor node localization. *Mobile Computing, IEEE Transactions on*, 4(6):567 – 577, nov.-dec. 2005.

[60] Guoqiang Mao, Baris Fidan, Guoqiang Mao, and Baris Fidan. *Localization Algorithms and Strategies for Wireless Sensor Networks*. Information Science Reference - Imprint of: IGI Publishing, Hershey, PA, 2009.

[61] David Tse and Pramod Viswanath. *Fundamentals of wireless communication*. Cambridge University Press, New York, NY, USA, 2005.

[62] R. H. Clarke. A statistical theory of mobile radio reception. *Bell Systems Technical Journal*, 47:957–1000, 1968.

[63] V. Erceg, P. Soma, D.S. Baum, and S. Catreux. Multiple-input multiple-output fixed wireless radio channel measurements and modeling using dual-polarized antennas at 2.5 ghz. *Wireless Communications, IEEE Transactions on*, 3(6):2288–2298, 2004.

[64] L. Thiele, M. Peter, and V. Jungnickel. Statistics of the ricean k-factor at 5.2 ghz in an urban macro-cell scenario. In *Personal, Indoor and Mobile Radio Communications, 2006 IEEE 17th International Symposium on*, pages 1–5, 2006.

[65] A. Abdi, C. Tepedelenlioglu, Mostafa Kaveh, and G. Giannakis. On the estimation of the k parameter for the rice fading distribution. *Communications Letters, IEEE*, 5(3):92–94, 2001.

[66] L.J. Greenstein, D.G. Michelson, and V. Erceg. Moment-method estimation of the ricean k-factor. *Communications Letters, IEEE*, 3(6):175–176, 1999.

[67] G. Azemi, B. Senadji, and B. Boashash. Ricean k-factor estimation in mobile communication systems. *Communications Letters, IEEE*, 8(10):617–619, 2004.

[68] A. Doukas and G. Kalivas. Rician k factor estimation for wireless communication systems. In *Wireless and Mobile Communications, 2006. ICWMC '06. International Conference on*, pages 69–69, 2006.

[69] Jr. William H. Hayt and John A. Buck. *Engineering electromagnetics*. Mc-Graw Hill, sixth edition, 2001.

[70] J. Tiusanen. Attenuation of a soil scout radio signal. *Biosystems Engineering*, 90(2):127 – 133, 2005.

[71] A. Doukas and G. Kalivas. Rician k factor estimation for wireless communication systems. In *Wireless and Mobile Communications, 2006. ICWMC '06. International Conference on*, page 69, july 2006.

[72] D.A.S Fraser and A.C.M. Wong. An approximation for the noncentral chi-squared distribution. In *Communications in Statistics - Simulation and Computation*, volume 27, pages 275–287, 1998.

[73] Carl W. Helstrom. *Statistical theory of signal detection*. Pergamon press, second edition, 1968.

[74] V. L. Mironov. Spectral dielectric properties of moist soils in the microwave band. In *Geoscience and Remote Sensing IEEE International Symposium*, 2004.

[75] M. Valery, K. Yann, W. Jean-Pierre, K. Liudmila, D. François, and D. Clement. Statistical error for the moistures retrieved with the smos radiobrightness data, as induced by imperfectness of a dielectric model used. In *Geoscience and Remote Sensing Symposium (IGARSS), 2010 IEEE International*, pages 4430–4432, 2010.

[76] V.L. Mironov, L.G. Kosolapova, and S.V. Fomin. Physically and mineralogically based spectroscopic dielectric model for moist soils. *Geoscience and Remote Sensing, IEEE Transactions on*, 47(7):2059–2070, 2009.

[77] N.R. Peplinski, F.T. Ulaby, and M.C. Dobson. Dielectric properties of soils in the 0.3-1.3-ghz range. *Geoscience and Remote Sensing, IEEE Transactions on*, 33(3):803–807, 1995.

[78] J.H. Scott and United States. Air Force. *Electrical and Magnetic Properties of Rock and Soil*. Open-file report. U.S. Geological Survey, 1983.

[79] Texas Instruments. Low-power soc (system-on-chip) with mcu, memory, sub-1 ghz rf transceiver, and usb controller. http://www.ti.com/lit/ds/symlink/cc1110f32.pdf.

[80] Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: Open source scientific tools for Python, 2001–.

[81] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing In Science & Engineering*, 9(3):90–95, 2007.

[82] J.D. Sanchez-Heredia, J.F. Valenzuela-Valdes, A.M. Martinez-Gonzalez, and D.A. Sanchez-Hernandez. Emulation of mimo rician-fading environments with mode-stirred reverberation chambers. *Antennas and Propagation, IEEE Transactions on*, 59(2):654–660, 2011.

[83] S.U. Rehman, T. Turletti, and W. Dabbous. Multicast video streaming over wifi networks: Impact of multipath fading and interference. In *Computers and Communications (ISCC), 2011 IEEE Symposium on*, pages 37–42, 2011.